Reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentations locales

Eric Nowak

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Mathématiques et Informatique

préparée au laboratoire GRAVIR – IMAG, projet LEAR,
dans le cadre de **l'Ecole Doctorale Mathématiques, Sciences et Technologie de l'Information**

présentée et soutenue publiquement

par

Eric NOWAK

le 17 Mars 2008

Reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentations locales

---

Directeur scientifique de thèse : Pr. Frédéric JURIE
Directeur industriel de thèse : Jean-Luc MAILLART
Co-Directeur scientifique de thèse : Pr. Roger MOHR

JURY

Pr. James CROWLEY,	Président
Pr. Nozha BOUJEMAA,	Rapporteur
Pr. Bernt SCHIELE,	Rapporteur
Pr. Frédéric JURIE,	Examinateur
M. Jean-Luc MAILLART,	Examinateur
Pr. Roger MOHR,	Examinateur

A QUELQU'UN

RECONNAISSANCE DE CATÉGORIES D'OBJETS ET D'INSTANCES D'OBJETS À L'AIDE DE REPRÉSENTATIONS LOCALES

Eric NOWAK, Ph.D. dissertation
Institut National Polytechnique de Grenoble, 17 Mars 2008

La reconnaissance d'objets est l'un des domaines d'étude les plus actifs de la vision par ordinateur. Il faut distinguer la reconnaissance de catégories d'objets génériques (une voiture en général, un piéton en général) et la reconnaissance d'instances d'objets particuliers (la voiture de M. Dupont, M. Dupont lui-même). Cette thèse aborde les deux sujets. Nous utilisons pour cela des représentations d'objets par parties, ce qui signifie que l'image à analyser n'est pas considérée dans son ensemble de manière rigide, mais plutôt comme un ensemble de régions locales, ce qui apporte une grande robustesse à la reconnaissance.

Nous nous intéressons spécifiquement à la reconnaissance d'objets décrits par sacs-de-mots. Cela signifie que les relations géométriques entre les régions locales décrivant une image sont ignorées. Nous étudions en particulier l'influence des différentes composantes de la classification d'images par sac-de-mots, et nous montrons que le facteur le plus influent est la quantité de régions locales sélectionnées, et pour cette raison nous proposons une sélection aléatoire et en grande quantité de régions locales dans les images à décrire.

Dans le contexte de la thèse CIFRE effectuée en partenariat avec l'INRIA et Bertin Technologies, nous analysons la performance des méthodes sac-de-mots pour la reconnaissance des véhicules militaires en imagerie infra-rouge. Nous montrons que les paramètres algorithmiques se comportent comme en imagerie visible. Nous effectuons aussi une étude des paramètres opérationnels, telle que la distance cible-caméra, et montrons que les paramètres sensibles sont les occultations et la présence de fond texturé quand les cibles sont détourées avec une faible précision.

Nous étudions aussi le compromis entre performance et temps de calcul, et proposons une méthode de sélection de primitives adaptées aux classifieurs hiérarchiques multi-classes, qui fournissent un meilleur compromis performance / temps de calcul que la selection de primitives pour classifieurs plats.

Les trois études précédentes traitent de la reconnaissance de catégories d'objets. Nous nous intéressons aussi à la reconnaissance d'instances d'objets, et proposons une mesure de similarité destinée à des instances d'objets jamais vus lors d'une phase d'apprentissage. Cette mesure est basée sur la quantification par des arbres extrêmement aléatoires de paires de régions locales correspondantes sélectionnées dans les deux images à comparer.

Toutes ces études sont validées par des expérimentations importantes sur des bases de données publiques, et nous obtenons à chaque fois des résultats aussi bons, sinon meilleurs, que ceux de l'état de l'art.

LOCAL FEATURE BASED OBJECT CATEGORIES AND OBJECT INSTANCES RECOGNITION

Eric NOWAK
Institut National Polytechnique de Grenoble, 17 Mars 2008

Object recognition is one of the most active fields of computer vision. In this thesis we consider two problems: recognition of object categories (a car, a pedestrian) and recognition of object instances (Mr Smith's car, Mr Smith himself). We use local object representations, which means that an image is considered as a set of local regions, which is more robust and more flexible that a global representation.

We particularly focus on bag-of-words methods, that discard geometric information between local regions. We study the influence of each step of the algorithm, and show that the parameter the most influent on the accuracy is the amount of local regions sampled to describe the image. We thus propose to sample a large amount of random local regions to describe images.

In the context of this CIFRE industrial PhD thesis, in partnership with INRIA and Bertin Technologies, we study how performant bag-of-words methods are for recognizing military vehicles on infrared images. We show that the algorithm parameters have the same behavior as the ones in the visible spectrum. We also study operation parameters, such as the distance between the camera and the target, and show that the most critical parameters are the occlusion rate and the amount of textured background in the region of interest when targets are poorly segmented.

We also study the trade-off between accuracy and computation time, and we propose a feature selection scheme well suited for multiclass hierarchical classifiers, more interesting than standard feature selection for flat classifiers.

The three previous studies focus on object category recognition. We also consider object instance recognition, and we propose a similarity measure for comparing objects never seen during a training phase. That measure is based on the quantization by extremely randomized clustering forests of matching pairs of local regions sampled from the two images to compare.

All these studies are validated by many experiments on state of the art and our own datasets, and we always obtain results as good as the state of the art, if not better.

REMERCIEMENTS

J’aimerais avant tout exprimer ma gratitude et mes remerciements à mon directeur scientifique de thèse, Frédéric Jurie. J’ai énormément appris à ses côtés. Il a su m’initier à la recherche, me pousser à toujours faire mieux, et m’a souvent aidé à surmonter les difficultés de ce cheminement qu’est le doctorat. Encore plus que ses grandes qualités scientifiques, j’ai beaucoup apprécié ses qualités humaines, en particulier l’écoute, la bonne humeur, le partage, la compréhension, et une grande complicité, qui m’en ont fait un ami.

J’adresse à mon directeur industriel de thèse, Jean-Luc Maillart, mes plus sincères remerciements. C’est un homme d’une grande intelligence, très à l’écoute et très compréhensif, et qui est directement responsable du bon déroulement de mes travaux. Je le remercie chaleureusement pour ses attentions et les nombreuses discussions professionnelles et personnelles que nous avons eues.

Je veux absolument remercier Cordelia Schmid, Bill Triggs et Roger Mohr, sans qui l’équipe LEAR ne serait pas ce qu’elle est aujourd’hui. Je remercie aussi le président du jury, James Crowley, et mes deux rapporteurs, Nozha Boujemaa et Bernt Schiele. Un énorme merci à ce dernier qui m’a invité dans son laboratoire à Darmstadt, et avec qui j’ai eu une discussion déterminante pour ma carrière.

"Big Hug" pour mes collègues de bureau INRIA et Bertin. Je pense en particulier à Gyorgy, Matthijs, Jianguo, Salil, Yves, Peter dont j’ai été (et suis toujours) très proche. Ils m’ont beaucoup apporté pendant ces dernières années: joie, fous rires, complicité, soirées, voyages, ouverture d’esprit, en un mot comme en mille une sincère amitié. Je dois aussi de bons moments de rigolade et complicité à Diane, Alexander, Anne, Hedi, Michael, Marcin, Ankur, Navneet, Matthieu, Joost, Guillaume, Hakan, Tijmen, Hervé, Jakob, Juliette, Benjamin, Xiaoyang, Emmanuel, Louis, Eric, Cédric, Marie, Chhorb.

Comme il y a aussi une vie en dehors du travail (si tout de même un petit peu) j’ai eu la chance de rencontrer hors du labo et de l’entreprise des personnes qui sont aujourd’hui des amis: Stan, Marlena, Augustin, Vanize, et tant d’autres! De plus, en optimisant le temps précieux passé hors de mon bureau, j’ai pu conserver et renforcer des amitiés précieuses, qu’elles datent de la maternelle ou de l’école d’ingénieur.

Enfin, je remercie ma famille pour m’avoir soutenu depuis 27 ans (et demi), et j’adresse un merci particulier à Audrey pour ce bout de chemin fait ensemble.

CONFIDENTIALITÉ

Cette thèse a bénéficié d'un financement CIFRE, de ce fait elle s'est déroulée au sein d'une entreprise: Bertin Technologies. Afin de préserver le savoir-faire acquis par Bertin Technologies, certaines données sensibles (telles que la performance sur certaines bases de données ou bien les réglages de certains algorithmes) doivent être confidentielles, et de ce fait ne sont pas publiées dans cette thèse.

Dans ce manuscrit, les données sensibles seront remplacées par *XX*. Les tableaux et figures sensibles seront supprimés.

Contents

1 Introduction **17**
- 1.1 La reconnaissance visuelle par ordinateur . . . 17
  - 1.1.1 But de la reconnaissance visuelle . . . 18
  - 1.1.2 Difficultés de la reconnaissance visuelle par ordinateur . . . 18
  - 1.1.3 Applications pratiques . . . 19
- 1.2 Principe d'un algorithme de reconnaissance visuelle . . . 20
  - 1.2.1 Re-connaissance . . . 20
  - 1.2.2 Les trois étapes fondamentales . . . 21
  - 1.2.3 Apprentissage machine . . . 22
- 1.3 État de l'art . . . 24
  - 1.3.1 Reconnaissance d'instance d'objets . . . 24
  - 1.3.2 Reconnaissance de catégories d'objets . . . 26
- 1.4 Organisation du document, contributions et publications . . . 30

2 Reconnaissance de catégories d'objets par sac-de-mots **33**
- 2.1 Résumé du chapitre . . . 33
- 2.2 Introduction . . . 34
- 2.3 Travaux apparentés . . . 35
- 2.4 Bases d'images . . . 37
  - 2.4.1 Bases d'images d'objets . . . 37
  - 2.4.2 Bases d'images de textures . . . 37
- 2.5 Paramètres expérimentaux . . . 38
- 2.6 Méthode de sélection de régions locales . . . 42
- 2.7 Vocabulaire Visuel . . . 45
  - 2.7.1 Taille du vocabulaire visuel . . . 45
  - 2.7.2 Algorithme de construction du vocabulaire visuel . . . 46
  - 2.7.3 Images utilisées pour la construction du vocabulaire visuel . . . 46
- 2.8 Normalisation des histogrammes . . . 47

2.9 Echelle minimum de sélection des régions locales . . . 48
2.10 Resultats sur les bases du challenge Pascal 2005 . . . 48
2.11 Conclusions . . . 49

3 Descriptions sacs-de-mots d'images infra-rouge **55**
3.1 Résumé du chapitre . . . 55
3.2 Introduction . . . 56
3.3 Base de données . . . 57
3.3.1 Méthode de génération . . . 58
3.3.2 Types d'objets . . . 58
3.3.3 Contenu . . . 59
3.4 Algorithme de classification . . . 59
3.4.1 Principe général . . . 60
3.4.2 Le descripteur d'images . . . 61
3.4.3 Classifieurs SVMs . . . 66
3.4.4 Conclusion . . . 67
3.5 Métrique d'évaluation de performance . . . 68
3.6 Étude des paramètres algorithmiques . . . 69
3.6.1 Paramètres algorithmiques . . . 69
3.6.2 Méthodes d'évaluation . . . 69
3.6.3 Hauteur standard de ROI, H . . . 70
3.6.4 Taille du vocabulaire visuel, D . . . 70
3.6.5 Taille de région locale standard, w_{std} . . . 75
3.6.6 Écart type maximum d'une région uniforme $\sigma_{uniform}$ . . . 75
3.6.7 Paramètres d'échantillonnage H_{max}, H_{min}, ds . . . 75
3.6.8 Offset de grille d'échantillonnage, off . . . 78
3.6.9 Seuils NCC d'activation $ncc_1, ncc_2, ..., ncc_k$ . . . 79
3.6.10 Quantité d'images nécessaires à l'apprentissage . . . 80
3.6.11 Compromis d'apprentissage du SVM . . . 82
3.6.12 Conclusion de l'étude algorithmique . . . 82
3.7 Étude des paramètres opérationnels . . . 83
3.7.1 Discrétisation des variables . . . 84
3.7.2 Configuration des paramètres algorithmiques . . . 85
3.7.3 Stabilité en situation de jamais-vu . . . 85
3.7.4 Stabilité en situation de déja-vu . . . 86
3.7.5 Nombre de classes . . . 87
3.7.6 Influence de σ_{atmo} . . . 88
3.7.7 Influence des configurations RSS/RSC . . . 88
3.7.8 Influence des conditions d'apprentissage . . . 89
3.7.9 Influence de la distance . . . 89
3.7.10 Conclusion de l'étude des paramètres opérationnels . . . 90
3.8 Informations pratiques . . . 91

3.8.1 Échantillonnage . . . 91
3.8.2 Temps de calcul . . . 92
3.8.3 Matrices de confusion . . . 93
3.9 Conclusion . . . 94
3.9.1 Étude de l'algorithme . . . 94
3.9.2 Étude des paramètres opérationnels . . . 95

4 Classification multi-classes: compromis temps-performance 97
4.1 Résumé du chapitre . . . 97
4.2 Introduction . . . 98
4.3 Représentation d'objets par parties locales . . . 99
4.3.1 Calcul d'un vocabulaire visuel . . . 99
4.3.2 Détection de parties locales . . . 100
4.3.3 Représentation d'images . . . 100
4.4 Classification multi-classes . . . 101
4.5 Sélection de primitives . . . 102
4.6 Méthode proposée . . . 105
4.6.1 Sélection de primitives pour hiérarchie de classifieurs binaires . . . 105
4.6.2 Transformation de données . . . 107
4.6.3 Algorithme de classification proposé . . . 108
4.7 Expérimentations . . . 110
4.7.1 Bases de données . . . 110
4.7.2 Mesure de performance . . . 114
4.7.3 Production du vocabulaire visuel . . . 115
4.7.4 Sélection de primitives . . . 115
4.7.5 Sélection de primitives pour hiérarchie de classifieurs binaires . . . 116
4.7.6 Normalisation des données . . . 118
4.7.7 Base infra-rouge: performance en situation de jamais-vu . . . 120
4.7.8 Observation de la sélection de primitives . . . 134
4.8 Conclusion . . . 136

5 Comparaison d'objets jamais vus 137
5.1 Résumé du chapitre . . . 137
5.2 Introduction . . . 138
5.2.1 Travaux similaires . . . 139
5.3 Mesure de similarité basée sur des parties locales . . . 141
5.3.1 Quantification des différences locales . . . 141
5.3.2 Vue d'ensemble . . . 142
5.3.3 Similarité entre deux images . . . 143
5.3.4 Apprentissage des arbres extrêmement aléatoires . . . 144
5.3.5 Des conditions booléennes multi-modales . . . 145
5.4 Résultats expérimentaux . . . 146

5.4.1 Bases de données . . . 146
5.4.2 Évaluation paramétrique . . . 147
5.4.3 Connaissances génériques et spécifiques . . . 153
5.4.4 Performance et comparaison avec l'état de l'art . . . 154
5.4.5 Combinaison des types de condition booléenne . . . 156
5.4.6 Visualisation des similarités . . . 159
5.4.7 Approches n'ayant pas abouti . . . 165
5.5 Discussion et conclusion . . . 171

6 Conclusion **173**
6.1 Principales contributions . . . 173
6.2 Démarche et rappel des résultats . . . 174
6.2.1 Intérêt pour la méthode sac-de-mots . . . 174
6.2.2 Etude des algorithmes sac-de-mots . . . 175
6.2.3 Algorithmes sac-de-mots et images infra-rouges . . . 175
6.2.4 Compromis temps de calcul / performance . . . 176
6.2.5 Identification d'objets jamais vus . . . 176
6.3 Perspectives . . . 177

FIRST CHAPTER

Introduction

1.1 La reconnaissance visuelle par ordinateur

Dans l'Allégorie de la Caverne [72], Platon présente sa théorie des Idées, et introduit les notions de Forme et de Chose. Les Formes sont des modèles, des concepts, et les Choses sont leurs réalisations. Par exemple, le cercle mathématique (le concept) est une Forme, et un cercle dessiné sur une feuille est une Chose, ce que l'on peut appeler une réalisation ou une instance du concept.

Dans le cadre plus moderne de la reconnaissance d'objets en vision par ordinateur, on différencie de la même manière les catégories d'objets (associées aux Formes) et les instances d'objets (associées aux Choses). Les bicyclettes, les ordinateurs, les visages, sont des exemples de catégories d'objets. La bicyclette rouge et verte de M. Dupont, un ordinateur Toshiba A-610 et le visage de M. Dupont sont des instances d'objets.

Un adulte peut reconnaître en général plus de 10 000 catégories d'objets [7], et une quantité bien plus importante d'instances d'objets. Le processus de reconnaissance est souvent rapide, sans effort, et robuste aux changements de point de vue, de luminosité, à des occultations d'une partie de l'objet, etc. L'apprentissage d'une nouvelle catégorie d'objets peut se faire à partir d'un petit nombre d'images de cet objet, et l'apprentissage d'une nouvelle instance d'objet peut se faire à partir d'une seule image de cet objet.

Il y a un grand intérêt à programmer des ordinateurs pour qu'ils soient eux aussi capables de reconnaître des catégories d'objets ou des instances d'objets de la même manière. Ce champ de la recherche se nomme *reconnaissance visuelle par ordinateur.* Après une phase d'apprentissage, où quelques images de la catégorie ou de l'instance d'objet sont fournies, l'ordinateur prédit si une image contient ou non cette catégorie d'objet (reconnaissance de catégorie d'objet) ou cet objet spécifique (reconnaissance d'instance d'objet). Après une brève présentation de la reconnaissace visuelle, nous mentionnerons les principales difficulités de cette tâche puis des exemples d'application.

1.1.1 But de la reconnaissance visuelle

La reconnaissance visuelle par ordinateur consiste à prédire à l'aide d'un algorithme quel objet se trouve dans une image. En raisonnant en termes d'entrées et sorties d'un algorithme, l'entrée de l'algorithme est une image, la sortie est la classe de l'objet (tâche de reconnaissance de catégorie d'objets) ou la référence d'un objet précis (tâche de reconnaissance d'instance d'objet). Si l'image contient plusieurs objets, la sortie de l'algorithme peut être une liste d'objets.

Cette tâche diffère de la localisation d'objet (aussi appelée détection d'objet) ou de l'appariement d'objet. La première consiste à déterminer précisément la position d'un objet dans une image, la seconde consiste à trouver les correspondances point à point entre deux objets présents dans deux images.

1.1.2 Difficultés de la reconnaissance visuelle par ordinateur

La première difficulté de la reconnaissance visuelle est de reconnaître des catégories d'objets qui ne sont pas caractérisées par une similarité visuelle. Si on considère la catégorie des véhicules, elle est avant tout fonctionnelle (le transport) et non visuelle. Il est tout a fait possible d'imaginer un véhicule d'une apparence si extravagante qu'un humain ne puisse le reconnaître comme tel. Dans cette thèse, nous ignorons donc les catégories fonctionnelles, et nous considérons plutôt la reconnaissance de catégories visuelles d'objets à partir d'images. Cette seconde tâche est plus simple dans l'état actuel de l'état de l'art, mais elle présente néanmoins d'importantes difficultés.

Si l'apparence des objets dans les images ne variait jamais, il serait très aisé de les reconnaître. Il suffirait de comparer pixel à pixel une région de l'image à une image de référence de catégorie connue pour savoir s'il s'agit du même objet (différence nulle) ou non (différence non nulle). Mais l'apparence des objets dans les images varie, et cette variation fait la difficulté de la tâche. Nous énumérons ci-dessous les causes de variation d'apparence les plus importantes.

Modification de point de vue. L'apparence d'un objet change considérablement quand le point de vue change, c'est à dire que l'objet subit une translation, rotation et changement d'échelle par rapport à la caméra. Il n'y a aucun point commun entre une voiture vue de face et de profil, et une personne n'ayant jamais vu de voiture penserait qu'il s'agit de deux objets différents si on lui montrait de telles photos (effet de rotation). De même, des indices visuels différents permettent de reconnaître un arbre vu de très près, dont on ne verrait qu'une branche et des feuilles, et un arbre vu a des kilomètres (effet d'échelle). Par contre, la translation, une fois décorrélée du changement de perspective, ne modifie pas l'apparence visuelle de l'objet.

Modification d'illumination. Les modifications d'illumination changent aussi l'apparence des objets. Une modification uniforme sur l'ensemble de l'image peut être annulée par une

rectification de la luminosité et du contraste à des valeurs standard, mais dans un cas réel, les changements ne sont pas uniformes, par exemple quand le soleil se déplace, quand des rideaux s'ouvrent, ou quand un lampe s'allume.

Déformation. Certains objets sont déformables, ce qui génère de grands changements d'apparence entre deux états. On peut considérer de petites apparitions/disparitions (comme une antenne radio rétractile sur une voiture), des déformations articulées, que subit un pantin articulé (chaque partie non articulée garde la même apparence mais l'apparence globale change) ou des déformations élastiques (visage qui fait une grimace).

Fond chargé. La position des objets dans les images est souvent déterminée par un rectangle englobant, aussi appelé Region Of Interest (ROI). Ce rectangle contient l'intégralité de l'objet, et aussi des pixels qui proviennent du fond de scène. Quand ce fond de scène change, le contenu du rectangle englobant change aussi, ce qui est une nouvelle source de variabilité.

Occultations. Les occultations de parties d'objets sont une grande source de variabilité de l'apparence des objets, car les apparences typiques de parties d'objets sont remplacées par l'objet occultant. Considérons un piéton qui marche derrière des arbres: chaque photographie prise de lui est différente en raison de la variation des formes de l'objet occultant, ce qui rend la reconnaissance très difficile.

Variations intra-classe. Les difficultés précédentes s'appliquent à la reconnaissance d'instances d'objets et de catégories d'objets. La variation intra-classe est spécifique aux catégories d'objets. Il s'agit du fait que plusieurs instances d'objets puissent avoir une apparence visuelle très différente, il suffit pour cela d'imaginer l'apparence des différents modèles de voitures. Quand un nouveau modèle de voiture est présenté à un algorithme, celui-ci doit faire le lien avec les autres modèles. Si la variation intra-classe est plus importante que la variation inter-classes (différences d'aspects entre instances d'objets de catégories différentes), alors la reconnaissance visuelle est impossible, le postulat de la reconnaissance visuelle étant que deux instances objets de catégories identiques aient une plus grande similarité[1] que deux instances d'objets de catégories différentes.

1.1.3 Applications pratiques

Nous listons dans cette section quelques applications pratiques de la reconnaissance visuelle de catégories et d'instances d'objets. Premièrement, on peut citer la recherche d'images dans des bases de données. On peut vouloir chercher dans sa collection personnelle toutes

[1] La tâche du chercheur en reconnaissance visuelle est, entre autre, de définir une telle mesure de similarité

les photos de personnes (catégories d'objet) ou de M. Dupont (instance d'objet). Il y a aussi les applications militaires et de sécurité: recherche d'individus ou de véhicules particuliers filmés par des milliers de caméras qui tournent 24 heures sur 24 dans une ville. Ces systèmes permettent de réduire la quantité d'humains qui observent les vidéos, qui posent des problèmes de coût et de vigilance. Une application récente de la reconnaissance visuelle est la conduite de véhicules autonomes, où la reconnaissance permet une conduite assistée (détection de piétons et voitures à proximité) ou totalement autonome (drône de repérage). On peut enfin citer la recherche d'images sur internet: aujourd'hui, cette recherche se fait uniquement à partir d'informations textuelles, comme le nom de l'image et le texte contenu dans la page où l'image apparaît. On peut imaginer un système qui demande quelques photos à l'utilisateur et qui recherche ensuite des images similaires.

1.2 Principe d'un algorithme de reconnaissance visuelle

Les algorithmes que nous présenterons dans l'état de l'art dans la section suivante ont des approches très différentes de la reconnaissance d'objets. Il existe cependant une trame commune à tous ces algorithmes, et nous la présentons dans cette section.

1.2.1 Re-connaissance

Le mot "reconnaître" se décompose naturellement en *re-* et *-connaître*. Dans un premier temps, il y a une phase d'acquisition de connaissance *(-connaître)*, il s'agit d'apprendre à caractériser l'objet qui nous intéresse. Qu'est ce qui fait que je peux reconnaître une voiture? Qu'est ce qui la différencie des autres objets? Il y a ensuite une phase d'utilisation de cette connaissance *(re-)*, qui analyse de nouveaux objets pour déterminer s'ils appartiennent à la catégorie apprise.

Les algorithmes de l'état de l'art fonctionnent avec la même logique. Ils ont une phase de caractérisation de l'objet d'intérêt, et une phase de recherche de cet objet. Nous illustrons notre propos à l'aide de deux approches très différentes de l'état de l'art, qui seront développées dans la section consacrée à l'état de l'art.

La première approche, proposée par Lowe en 1987 [54] est une approche par alignement géométrique. Elle est destinée à reconnaître les instances d'objets. Un objet est représenté par un modèle 3D, et il est reconnu dans les images en alignant les droites du modèle aux droites détectées dans l'image. Si l'alignement est correct, on considère que l'objet est détecté. La phase d'acquisition de connaissances consiste ici à fournir à l'algorithme un modèle 3D de l'objet recherché. Une fois que l'algorithme dispose de ces connaissances nécessaires, l'objet peut être recherché dans des images. Pour ce faire, des droites sont détectées dans l'image, et des appariements sont calculés entre les droites du modèle et les droites de l'image. Une mesure de similarité permet de mesurer à quel point le modèle et l'image correspondent, si cette mesure est suffisamment élevée on considère que l'objet est reconnu.

La seconde approche, beaucoup plus moderne, a été proposée par Dalal et Triggs en 2005 [17]. Elle est destinée à reconnaître des catégories d'objets, des piétons en l'occurrence. Une région contenant un piéton est représentée par la distribution de l'orientation des gradients dans cette région. Cela permet d'encoder une information comme: "en haut à droite de la région, il y a beaucoup de gradients horizontaux." La phase d'acquisition de connaissance consiste à caractériser la répartition des gradients dans les régions contenant des piétons. Lorsqu'une nouvelle région doit être analysée, les gradients sont calculés, et une mesure de similarité mesure à quels points ceux-ci correspondent aux gradients typiques que l'on observe sur des piétons. A la différence de l'approche précédente, les caractéristiques des piétons ne sont pas spécifiées manuellement, mais apprises automatiquement par l'algorithme. L'algorithme observe des images positives (contenant des piétons) et des images négatives (ne contenant pas de piéton) et détermine automatiquement ce qui permet de les différencier au mieux. Le champ de recherche qui étudie l'apprentissage automatique se nomme *apprentissage machine*, et nous en dirons quelques mots ci-dessous.

1.2.2 Les trois étapes fondamentales

Les algorithmes de reconnaissance d'instances ou de catégories d'objets sont basés sur trois composantes fondamentales.

La première, mentionnée explicitement ci-dessus, est une phase *d'acquisition de connaissance*, ou d'apprentissage d'un modèle. Elle peut se faire explicitement, comme dans le premier exemple, ou implicitement et automatiquement, comme dans le second. Les approches modernes privilégient la seconde méthode, car une modélisation explicite d'un objet complexe nécessite une grande expertise si l'on veut considérer tous les cas de figure.

La seconde composante apparaît en filigrane dans les deux descriptions de méthodes ci-dessus. Il s'agit de la notion de *mesure de similarité*, qui vient mesurer à quel point ce que l'on observe dans une image correspond au modèle considéré.

Enfin, la troisième composante fondamentale est la *transformation des images*. Dans la première méthode, le contenu originel de l'image (les pixels) est ignoré au profit des droites contenues dans l'image. Dans la seconde méthode, les pixels sont ignorés au profit des gradients. Ces transformations d'images sont motivées par le choix du modèle: un modèle à base de gradients aura besoin d'informations de gradient, et un modèle à base de droites aura besoin de détecter les droites de l'image. Le choix du modèle est lui même orienté par les trois considérations suivantes. *(a) La catégorie d'objet concernée.* Quels sont les éléments caractéristiques de la catégorie observée? S'il s'agit de bouteilles, la forme est un élément caractéristique, s'il s'agit d'arbres, alors la texture sera plus caractéristique. *(b) Les variabilités à considérer.* En fonction des transformations que peuvent subir les objets, des primitives différentes pourront être considérées. Sur une chaîne de traitement industrielle aux conditions d'éclairage constantes, les niveaux de gris de l'image suffisent. Si l'éclairage varie non uniformément, les gradients peuvent être plus utiles. Si l'objet peut être occulté, les représentations globales (détaillées plus loin) devront être évitées. *(c) Les contraintes informatiques.* S'il y a des contraintes sur le temps de calcul ou l'utilisation de la mémoire,

les primitives et algorithmes économiques en temps de calcul et/ou ressources memoires seront favorisées.

1.2.3 Apprentissage machine

Nous l'avons mentionné ci-dessus, les connaissances peuvent être intégrées par un expert ou bien apprises automatiquement par l'algorithme, ce que l'on nomme apprentissage machine. On distingue deux grandes familles d'apprentissage: l'apprentissage *génératif* et l'apprentissage *discriminatif*. Le premier apprend un modèle dans l'absolu, et de ce fait dispose des règles pour *générer* de nouveaux exemples. Par exemple: un visage contient deux yeux à telle position, une bouche à telle position, les yeux peuvent ressembler à ceci, la bouche peut ressembler à cela, etc. Le second apprend ce qui différencie un modèle des autres modèles. Par exemple, pour différencier un visage d'une échelle, nul besoin de savoir à quoi ressemblent un visage ou une échelle, il suffit par exemple de seuiller le nombre de droites parallèles détectées dans l'image (valeur élevée pour une échelle), car cette caractéristique *discrimine* les deux catégories.

Introduisons maintenant le formalisme nécessaire à la présentation de l'apprentissage génératif et discriminatif. Notons y_i la catégorie de l'objet numéro i contenu dans une base d'images d'apprentissage, dont les caractéristiques sont mesurées par la variable multidimensionnelle x_i. Les catégories sont représentées par des numéros, s'il y a C catégories alors $y_i \in \{1,\ldots,C\}$. L'apprentissage machine se fait à partir de n exemples d'apprentissage $\{(x_1,y_1),\ldots,(x_n,y_n)\}$ et son but est la prédiction de la catégorie inconnue y d'une nouvelle image dont les caractéristiques mesurées sont x. Pour prendre un exemple concret, y peut valoir 1 pour signifier "visage" et 0 pour signifier "pas de visage", et x peut être la concaténation en un vecteur des pixels d'une région de taille 20x20.

Apprentissage génératif

L'apprentissage génératif modélise la probabilité jointe des données (x) et des catégories (y) c'est à dire $P(x,y)$. La probabilité d'une catégorie sachant les données $P(y=k|x)$ est obtenue par marginalisation et avec la loi de Bayes. La classe prédite est la classe la plus probable, soit celle qui maximise $\arg\max_{k=1,\ldots,C} P(y=k|x)$. La loi de Bayes permet d'obtenir cette probabilité a posteriori (après observation des données) à partir des probabilités à priori et de la vraisemblance, définis respectivement par $P(y=k)$ et $P(x|y=k)$. La difficulté de la tâche est donc de modéliser la distribution des données dans chaque classe, c'est à dire $P(x|y=k)$.

De nombreuses méthodes ont été proposées pour estimer la distribution des données d'une classe. Nous référons le lecteur à l'excellent ouvrage de C. Bishop [8] pour plus de détails sur les techniques d'estimation de densité.

Apprentissage discriminatif

Contrairement aux méthodes génératives, les méthodes discriminatives ne modélisent pas la distribution des données, mais modélisent directement la frontière de décision entre les catégories. C'est à dire qu'elles affectent à chaque point de l'espace des caractéristiques un score de confiance dans le fait qu'un point appartienne à telle catégorie plutôt qu'à une autre. Une fois de plus, nous référons le lecteur à l'ouvrage de C. Bishop [8] pour une présentation détaillée des différentes méthodes.

Nous évoquons tout de même ci-après les classifieurs SVM (Support Vector Machines, ou Machines à Vecteurs Support) [88], car ce sont ceux que nous utilisons principalement dans ce document. Dans le cadre de la classification binaire, le SVM est un algorithme qui apprend à séparer deux classes, notées par convention "+1" et "-1". L'algorithme produit un score de confiance $f(x)$ qui est d'autant plus élevé que l'algorithme considère que x est membre de la classe "+1". Notons que pour prendre une décision, il faut seuiller cette valeur de confiance. Au dessus du seuil, la donnée est prédite comme membre de la classe "+1", en deçà, elle est prédite comme membre de la classe "-1".

Le SVM calcule l'hyperplan dans l'espace des caractéristiques où se trouve x qui sépare au mieux les classes "+1" et "-1". Pour ce faire, il va considérer la marge entre la frontière et les données. Une marge faible signifie que la frontière passe à proximité des données, ce qui peut donner des erreurs de classification si les données sont bruitées, et donc que leur position mesurée dans l'espace des caractéristiqes n'est pas exactement leur position réelle. Si la marge est élevée et que le bruit dans les données est relativement faible, cette imprécision est sans conséquence sur la prédiction de catégorie.

Surapprentissage

L'apprentissage génératif et discriminatif sont tous deux susceptibles de faire du surapprentissage. Le but de l'apprentissage est de déterminer la catégorie des images qui n'ont jamais été vues lors de l'apprentissage. Un modèle en situation de surapprentissage, au contraire, a appris très exactement les caractéristiques des données d'apprentissage, au détriment de sa capacité à généraliser sur des données inconnues. La généralisation est le principe de base de l'inférence, qui permet de passer d'un ensemble d'exemples (des images de voitures) à un modèle (le modèle des voitures), ou pour reprendre les mots de Platon [72], des Choses aux Formes.

Le surapprentissage intervient en particulier quand le modèle choisi dispose de trop de paramètres par rapport à la quantité de données disponibles. Comme les SVMs intègrent par construction une procédure de sélection de primitives, ils limitent automatiquement la quantité de paramètres ce qui réduit (sans les annuler totalement) les risques de surapprentissage.

La performance de classification sur les données d'apprentissage se nomme risque empirique, et cela correspond à la quantité d'erreurs effectuées sur les données d'apprentissage. Ce score peut être précisément évalué grâce aux données d'apprentissage. La performance

sur l'ensemble des données possibles se nomme risque réel, il s'agit de la performance absolue de l'algorithme. Celle-ci ne peut être évaluée en pratique. En situation de surapprentissage, le risque empirique est très faible car l'algorithme apprend précisément les données d'apprentissage, par contre le risque réel est élevé, car l'algorithme ne sait pas généraliser aux données inconnues. Quand le surapprentissage est évité, le risque empirique est plus élevé qu'en situation de surapprentissage (car l'algorithme ne colle pas aux données) mais le risque réel est plus bas. Notons que le risque réel peut être évalué de manière approximative avec des données de test, c'est à dire un ensemble de données différentes des données d'apprentissage mais générées par le même phénomène.

Niveau de supervision

La supervision est la quantité d'information associée aux données observées. On distingue en général trois niveaux. L'apprentissage supervisé considère que la catégorie y_i de chaque donnée d'apprentissage x_i est connue. Dans le contexte de notre thèse, cela signifie que l'objet présent dans chaque image d'apprentissage est connu, et que l'algorithme peut utiliser cette information. L'apprentissage semi-supervisé considère que la catégorie y_i des données d'apprentissage est connue pour une faible quantité de données, et inconnue pour une grande majorité des données. Enfin, l'apprentissage non supervisé considère que seules les données sont disponibles, et que les catégories associées sont inconnues. Dans ce dernier cas, les modèles selectionnés sont ceux qui maximisent la vraissemblance des données.

1.3 État de l'art

Nous présentons successivement les méthodes de l'état de l'art concernant la reconnaissance visuelle par ordinateur d'instance d'objets puis de catégories d'objets.

1.3.1 Reconnaissance d'instance d'objets

Le but de ces méthodes est de déterminer si une image contient des objets de référence, et éventuellement la position de ces objets dans l'image. Pour ce faire, les objets de référence sont représentés par un modèle 3D (méthodes géométriques) ou par des bases de données d'images de référence (méthodes globales et méthodes locales).

Méthodes géométriques

Les premières méthodes de reconnaissance d'instances d'objet sont des méthodes géométriques [41, 54, 37, 40, 84, 85, 29, 46, 95]. Elles consistent à représenter les objets de référence par leurs contours. On distingue les méthodes à base d'alignement modèle-objet et les méthodes à base de tables de hachage.

Les méthodes d'alignement géométrique disposent d'une modèle 3D de l'objet recherché. Les primitives (par exemple des droites) composant ce modèle 3D sont alignées sur

les primitives détectées dans l'image, et la qualité de l'alignement détermine si l'objet est présent ou non [41, 54]. Le calcul des correspondances entre les primitives du modèle et de l'image est un processus NP-complet, des heuristiques sont donc nécessaires pour réduire la complexité du problème. On peut citer les arbres d'interprétation [37, 40], qui sont des arbres de tous les alignements modèle-image possibles, élagués par l'application d'heuristiques. Par ailleurs, Fischler et al [29] appliquent la méthode RANSAC pour obtenir un consensus d'alignement à partir d'un sous ensemble des primitives de l'image, et décident de la validité de l'alignement avec les primitives de l'ensemble complémentaire.

Lamdan [46] et Wolfson [95] utilisent des tables de hachage pour la reconnaissance. Les primitives géométriques des différents modèles d'objets sont stockées dans une table de hachage, et lors de la reconnaissance, les primitives observées dans l'image sont recherchées dans la table de hachage, et des votes sont accumulés pour les différents modèles. Un objet est reconnu si le modèle correspondant obtient une quantité minimale de votes.

En ne considérant que le contour des objets, et donc en ignorant l'intérieur de ceux-ci, les méthodes géométriques obtiennent une grande robustesse aux changements d'illumination et de texture des objets. De plus, l'utilisation d'un modèle 3D permet de traiter les occultations et le fond chargé de manière très élégante. Ces méthodes souffrent cependant de deux défauts. Premièrement, certaines catégories d'objets, comme les arbres, ne peuvent être définis uniquement par leurs contours. Deuxièmement, ces méthodes ne sont pas robustes à une détection imprécise des contours, ce qui est souvent le cas dans des conditions d'acquisition d'images réelles.

Méthodes globales

Les méthodes de reconnaissance globale [87, 65, 73, 67, 79] calculent une signature de l'image prise dans sa globalité. Cette signature peut être, par exemple, la distribution des couleurs dans l'image. Une telle signature est calculée pour un ensemble d'images représentatives d'une instance d'objet. Lors de la reconnaissance, la signature de l'image inconnue est calculée, puis est comparée aux images de références. L'objet prédit est l'objet de référence le plus similaire à la signature de l'image inconnue.

La signature de l'image peut être constituée de l'ensemble des pixels mis les uns à la suite des autres dans un vecteur colonne. La comparaison avec les images de référence peut se faire avec une analyse en composante principale, ce qui est le cas des *eigenfaces* de Turk et Pentland [87], ou bien à l'aide d'une machine à vecteur support [88] comme le font Pontil et al [73]. Les signatures utilisées par Niblack et al [67] et Schiele et al [79] sont respectivement des histogrammes de couleurs et de textures.

Les méthodes globales sont très simples, et elles sont robustes aux modifications d'illumination et de contraste à condition que la base d'images de référence contiennent une grande quantité d'images aux conditions d'illumination différentes. Pour cette raison, les méthodes globales souffrent d'un besoin d'une grande quantité de données. De plus, la signature intégrant l'influence de tous les pixels de l'image, ces méthodes obtiennent de mauvais résultats en cas d'occultation et de présence de fond chargé.

Méthodes locales

Les méthodes locales [80, 53] pallient les principaux défauts des méthodes globales. Elles ne considèrent plus les images comme un tout monolithique, mais comme une collection de régions locales, qui sont le plus souvent des parties d'images carrées ou rectangulaires, pouvant contenir une centaine de pixels ou bien plus. Ces régions locales s'affranchissent des difficultés rencontrées sur des images entières, car elles peuvent être ramenées à une apparence standard. Les modifications d'illumination sont localement uniformes, et la luminosité et le contraste peuvent donc être normalisés. Ces régions locales correspondent généralement à des surfaces planes, donc leur orientation et leur échelle peuvent être ramenées à des valeurs standard [80]. Quand ces régions sont découpées à l'intérieur des objets, elles ne subissent pas d'influence du fond chargé, et en sont donc indépendantes. Enfin, quand ces régions sont de petites tailles, elles sont le plus souvent présentes intégralement ou bien totalement occultées, mais elles sont rarement partiellement occultées, et donc leur apparence ne varie pas à cause des occultations.

Le principe de ces méthodes est de représenter les images de référence par des régions locales et de stocker les descripteurs de ces régions locales (par exemple leurs niveaux de gris) dans une base de données. Lors de la reconnaissance, les descripteurs des régions locales sélectionnées dans l'image sont recherchés dans la base de données, ce qui permet de voter pour les images de référence qui contiennent des descripteurs similaires.

Schmid et al [80] utilisent des détecteurs de points d'intérêt pour obtenir des régions locales d'une orientation et échelle normalisées. Des contraintes géométriques sur les positions relatives des régions locales permettent de réduire le taux de fausse alarme. Lowe [53] propose un système temps réel et robuste pour rechercher les objets similaires dans une base de données, et introduit le fameux descripteur SIFT, qui encode la distribution de l'orientation du gradient dans une région locale. Dans ce cas aussi, la présence des objets est vérifiée à l'aide de contraintes géométriques sur les positions relatives des régions locales.

En raison de leur rapidité, de leur bonne gestion des occultations, du fond chargé et des modifications de point de vue, les méthodes locales ont aujourd'hui la préférence de la communauté pour la reconnaissance visuelle d'instances d'objets.

1.3.2 Reconnaissance de catégories d'objets

Considérons maintenant les méthodes de reconnaissance de catégories d'objets. Nous verrons que certains concepts introduits pour les instances d'objets, en particulier les représentation globales et les représentations locales, sont aussi à la base des méthodes de reconnaissance de catégories d'objets. Nous rappelons que les difficultés rencontrées pour la reconnaissance de catégories sont les mêmes que celles de la reconnaissance d'instances d'objets, auxquelles s'ajoute la variabilité intra-classe.

Méthodes globales

Les méthodes globales présentées ci-dessus pour la reconnaissance d'instances d'objets [87, 65, 73, 67, 79] peuvent être utilisées de la même manière pour la reconnaissance de catégories d'objets. En effet, ces méthodes consistent à rechercher l'objet le plus similaire dans une base de données, après transformation de l'image (histogramme de couleurs par exemple). Quand la variation intra-classe est faible, et que les variations inter-classes sont importantes, un objet est plus similaires aux objets de la même catégorie qu'aux objets des catégories différentes, ce qui rend possible la recherche par similarité dans une base de données.

Ces techniques ayant déjà été présentées plus haut, nous n'en parlerons pas davantage, si ce n'est pour indiquer qu'elles souffrent des mêmes inconvénients.

Méthodes locales

Pour les catégories d'images comme pour les instances d'images, des méthodes locales (prenant en compte des régions locales dans les images) ont été proposées pour contrer le principal inconvénient des méthodes globales: le manque de robustesse aux modifications locales (occultations, fond, illumination). Ces méthodes sont en général divisées en trois catégories: régions locales à géométrie figée, régions locales sans contraintes géométriques, et régions locales à la géométrie flexible. Nous les présentons ci-dessous.

Régions locales à la géométrie figée Ces méthodes considèrent en général qu'un objet a une taille standard et qu'il est fait de différentes parties dont les positions sont fixes. Par exemple, un visage doit tenir dans une région de 20x20 pixels, et le nez doit se trouver au centre et faire une taille de 5x5 pixels. Ces algorithmes prédisent si une région de taille standard (20x20 dans cet exemple) contient un objet de la catégorie recherchée (un visage dans cet exemple). Afin de déterminer la présence et la position des objets dans l'image, cet algorithme parcourt toutes les positions dans l'image, et il parcourt des versions sous-échantillonnées de l'image (formant une pyramide d'échelles) afin de détecter la présence de l'objet quelque soit sa taille.

Les travaux de Schneiderman et Kanade [81], Viola et Jones [90], Torralba et al [86] et Dalal et Triggs [17] font référence dans ce domaine. Nous n'expliquons ci-dessous que les algorithmes qui déterminent si une région d'intérêt (de taille 20x20 dans l'exemple ci-dessus) contient la catégorie recherchée ou non, la procédure de parcours multi-positions et multi-échelles de l'image totale pour retrouver les objets ayant déjà été expliquée ci-dessus.

Dans l'approche de Schneiderman et al [81], les parties locales proviennent d'une transformation en ondelettes de la région d'intérêt. Les parties ont une signification géométrique (pixels proches dans l'image d'origine) mais aussi une signification fréquentielle (fréquences proches dans la décomposition en ondelettes). L'algorithme détermine si l'objet est présent ou non en seuillant le rapport de vraisemblance, c'est à dire le ratio des probabil-

ités d'avoir un objet ou du fond sachant les parties observées, qui sont considérées indépendantes pour simplifier les calculs.

Viola et Jones [90] proposent une cascade de classifieurs utilisant des primitives simples et rapides à calculer, ce qui correspond au premier détecteur de visages robuste et temps réel. Les primitives utilisées sont des approximations de filtres dérivées de gaussiennes, qui se calculent très rapidement quelque soit la taille et la position des régions observées grâce à des images intégrales, issues de la programmation quadratique. Des classifieurs sont appliqués en cascade (en série) pour prédire si une région d'intérêt contient un visage ou non. Dès qu'une région d'intérêt est rejetée par un niveau de la cascade, elle est prédite comme négative (sans visage), ce qui permet de rejeter rapidement la majorité des régions de l'image. Seules les régions qui traversent avec succès tous les niveaux de la cascade sont prédites comme positives (avec visage). Les classifieurs situés à chaque niveau de la cascade sont inspirés du boosting [34].

Torralba et al [86] s'intéressent à la détection multi-classes, ce qui signifie que les positions d'objets de nombreuses catégories sont recherchées simultanément dans les images. Les primitives utilisées sont simples: les niveaux de gris des régions locales ainsi que leur position dans la région d'intérêt. La force de leur algorithme est le partage des primitives entre différentes classes, c'est à dire que les mêmes primitives sont utilisées pour déterminer la présence de multiples catégories. Cela permet d'obtenir un temps de calcul logarithmique en fonction du nombre de catégories, contrairement à un temps linéaire pour une approche basique qui testerait la présence des catégories l'une après l'autre.

Enfin, Dalal et Triggs [17] s'intéressent eux aussi à la reconnaissance de piétons. Ils calculent l'orientation du gradient dans chaque cellule d'une grille placée sur la région d'intérêt, et forment un vecteur à partir de ces statistiques. Un classifieur SVM [88] est appris à partir des descripteurs d'images avec et sans piétons de la base d'apprentissage. Ce classifieur est utilisé pour déterminer si le descripteur d'une nouvelle région d'intérêt représente un piéton ou non.

Ces méthodes locales à géométrie fixes sont bien plus robustes que les méthodes globales. Cependant, étant donné la rigidités des contraintes géométriques, elles nécessitent une très grande quantité d'images d'apprentissage afin de considérer tous les cas de figure possibles, comme les occultations de telle ou telle partie, ou la présence de fonds typiques.

Régions locales sans contraintes géométriques Cette méthode a été proposée par Csurka et al [16], et Opelt et al [71] et Zhang et al [98] en ont proposé des variantes. Cette approche étant à la base de nos travaux, nous la détaillerons amplement dans le chapitre 2, et en donnons ci-après un bref aperçu.

L'algorithme considère qu'une image est faite d'un grand ensemble de régions locales dont les positions sont ignorées. Des régions locales caractéristiques (i.e. d'apparence typique) sont déterminées et leur ensemble forme un *vocabulaire visuel*. Les régions locales sélectionnées dans les images sont comparées et associées aux éléments du vocabulaire visuel, ce qui permet d'obtenir la distribution des éléments du vocabulaire visuel dans les

images considérées. Ces distributions caractérisent les images, et sont utilisées par des classifieurs pour déterminer la présence ou l'absence d'une catégorie d'objet dans une image.

Cette méthode présente des avantages considérables par rapport aux autres méthodes de l'état de l'art. C'est une méthode très simple, ce qui réduit les risques de surapprentissage et limite la quantité de données nécessaires à l'apprentissage. De plus, lors de l'apprentissage, il n'est pas nécessaire d'indiquer la position de l'objet dans l'image, il suffit de savoir si l'image contient l'objet ou non. L'absence de contraintes géométriques et l'apprentissage statistique de la répartition des éléments du vocabulaire visuel la rendent très robuste aux occultations et à la présence de fond chargé. Cette méthode a fait ses preuves car elle obtient aujourd'hui les meilleures performances de l'état de l'art [19, 20]. Elle a cependant une faiblesse: l'absence de contraintes géométriques ne lui permet pas de localiser les objets une fois leur présence dans les images prédite.

Régions locales à la géométrie flexible Les méthodes par représentation locale à géométrie flexible modélisent à la fois l'apparence des régions locales et leurs positions possibles dans l'image. Elles sont inspirées des travaux de Fischler et Elschlager [28] sur les modèle de structure d'image, qui datent de 1973. Nous présentons ci-dessous les grandes familles de méthodes qui existent dans l'état de l'art.

Agarwal et al [3] modélisent l'apparence et la configuration géométrique des régions locales dans un vecteur descripteur. Les régions locales sont représentées par l'élément le plus proche d'un vocabulaire visuel précalculé, et les relations géométriques entre toutes les paires de mots visuels (éléments du vocabulaire visuel) détectés dans l'image sont encodées dans un vecteur descripteur, grâce à une quantification des relations géométriques (distance et orientation) entre régions locales. Un classifieur est appris pour séparer les images positives (contenant l'objet cherché) et négatives (ne le contenant pas).

Leibe et Schiele [49] proposent un modèle de forme implicite. Lors de l'apprentissage, des masques de segmentation précis des objets sont fournis, et chaque région locale quantifiée (i.e. représentée par un mot visuel) apprend la position du centre de l'objet par rapport à elle. Lors de la reconnaissance, les régions locales fournies par un détecteur d'intérêt sont à leur tour quantifiées, et chacune vote pour les positions possibles du centre de l'objet à reconnaître, ainsi que son échelle. Les positions et échelles accumulant une grande quantité de votes sont analysées lors d'un stage de vérification pour valider s'ils correspondent à des objets de la catégorie recherchée ou non.

L'approche de Fischler et al [28] a été successivement améliorée plus de quinze ans plus tard par Burl et al [12], Weber et al [91], Fergus et al [27] puis Fei-Fei et al [22]. Ils ont étendu les travaux de Fischler et ont proposé un modèle d'objet par constellation et un apprentissage peu ou pas supervisé. Le modèle consiste en un ensemble de parties typiques dont les positions sont similaires d'une instance d'objet à l'autre au sein de la même catégorie. Tout est modélisé explicitement dans un cadre probabiliste, de l'apparence des parties à leurs relations géométriques, en passant par l'existence de régions locales occultées ou la confusion entre fond et parties d'objets. Fergus parvient à s'affranchir des annotations lour-

des lors de l'apprentissage, il suffit à son approche de savoir si une image contient un objet ou non, et n'a donc pas besoin de la position précise de l'objet dans l'image. Cependant, son approche ne traite que des objets observés selon le même point de vue (pour garder une consistance géométrique), et sa complexité algorithmique est très grande car elles considère les relations entre toutes les paires de parties locales. Fei-Fei a montré que les modèles appris sur certaines catégories apportent de l'information sur de nouvelles catégories, ce qui lui permet dans certains cas d'apprendre de nouvelles catégories à partir d'une seule image.

Berg et al [5] considèrent la reconnaissance de catégorie d'objet comme un problème de correspondance entre un modèle déformable et une image. Un coût de déformation du modèle est évalué lors de la correspondance, et si ce coût est suffisamment faible, la catégorie correspondant au modèle est prédite. Cette méthode permet une estimation précise de la pose, et permet de gérer dans une certaine mesure les occultations et la présence de fond chargé. Par contre, elle ne permet de traiter que des objets dont les formes sont caractéristiques, et ne considère qu'un seul point de vue d'observation des objets.

Enfin, on peut citer les travaux de Felzenszwalb et al [23] et Ramanan et al [76] sur la reconnaissance de pose d'humains avec des méthodes elles aussi inspirées de Fischler et al [28].

1.4 Organisation du document, contributions et publications

Nous avons discuté de la reconnaissance visuelle de catégorie et d'instance d'objets en général dans ce chapitre, et avons en particulier fait un bilan de l'état de l'art dans la section 1.3.

Dans le chapitre 2, nous présentons en détails l'algorithme de reconnaissance de catégories d'objets par sac-de-mots. Il a été proposé par Csurka et al [16] en 2004, et de nombreuses équipes l'ont utilisé depuis, sans pour autant répondre à des questions fondamentales sur cet algorithme. Nous avons donc analysé les différentes étapes de l'algorithme, et mesuré l'importance relative des choix algorithmiques et paramétriques possibles. En particulier, nous montrons que lors de l'étape qui consiste à déterminer quelles zones de l'image doivent être considérées, il est préférable de sélectionner une grande quantité de régions aléatoirement plutôt que d'utiliser un détecteur de points d'intérêt. Nos validons nos conclusions sur trois bases d'images publiques de catégories d'objets et trois bases d'images publiques de textures. Ces travaux ont donné lieu à une publication à ECCV en 2006 [70].

Le chapitre 3 considère le problème de la reconnaissance visuelle d'objets en imagerie infrarouge par sac-de-mots. Ce chapitre est motivé par le contexte de la thèse CIFRE, dont l'un des objectifs est la reconnaissance de véhicules militaires en imagerie infrarouge. Nous vérifions donc les conclusions du chapitre précédent sur l'influence des différents composants de l'algorithme, et menons une étude sur les paramètres algorithmiques (tels que la taille du vocabulaire visuel) et opérationnels (tels que la distance caméra-cible et le contraste cible-fond). Nous montrons que l'algorithme donne de très bons résultats et que le réglage des paramètres algorithmiques n'est pas sensible. Nous montrons aussi que les

paramètres opérationnels les plus sensibles sont l'occultation des véhicules et la présence de fonds très texturés. Ces travaux se sont concrétisés sous la forme de rapports fournis à Bertin Technologies et à la DGA.

Le chapitre 4 s'intéresse à la diminution du temps de calcul pour prédire la catégorie d'une image. Cette section s'intéresse donc au compromis temps de calcul / performance, dans le cas de la classification multi-classes. La solution que nous proposons consiste à faire de la sélection de primitives visuelles combinée à un classifieur hiérarchique, ce qui permet de ne détecter à chaque étape du processus de classification multi-classes que les primitives visuelles les plus utiles. Notre approche est validée sur une base de véhicules en imagerie infrarouge et sur une base d'objets en imagerie visible. Ces travaux ont donné lieu à une publication à VS-PETS en 2005 [68].

Enfin, le chapitre 5 propose un algorithme de reconnaissance d'instance d'objets (ou identification d'objet) qui n'a pas les défauts de la représentation par sac-de-mots. En effet, la représentation sac-de-mots est très adaptée à la reconnaissance de catégories d'objets, mais manque de précision pour la reconnaissance d'instances d'objets, pour laquelle les petits détails (comme l'apparence d'une poignée de porte de voiture) ont une grande importance. Nous proposons donc une mesure de similarité visuelle entre deux images, basée sur des arbres de classification extrêmement aléatoires. Cette mesure donne un score de similarité entre deux images, même si les objets représentés sur ces images n'ont jamais été vus lors d'une phase d'apprentissage. Cette mesure, une fois seuillé, permet de prédire si les objets sont similaires ou différents. Nous obtenons d'excellents résultats sur notre base de données de voitures jouets et sur les bases de données publiques des auteurs traitant la même problématique. Ces travaux ont donné lieu à une publication à CVPR en 2007 [69], ainsi qu'une publication dans PAMI en 2007 [63].

SECOND CHAPTER

Reconnaissance de catégories d'objets par sac-de-mots

2.1 Résumé du chapitre

En raison de leur simplicité et de leur grande performance, les représentations d'images par sac-de-mots sont désormais très populaires. Elles sont la suite logique des méthodes d'analyse de texture par textons. La philosophie de cette représentation consiste à traiter une image comme un ensemble de régions indépendantes. Pour décrire une image, il faut sélectionner (ou échantillonner) un ensemble de régions locales significatives, décrire chacune d'elles avec un descripteur visuel local, et caractériser l'image par la distribution de ses descripteurs visuels. Les quatre étapes algorithmiques suivantes sont donc primordiales. Comment sélectionner les régions locales? Comment les décrire? Comment caractériser leur distribution? Comment utiliser ces caractérisitiques pour classifier une image? Nous nous interessons ici à la première question, à laquelle aucune étude systematique n'a répondu auparavant, et nous montrons expérimentalement que pour des bases de données d'images publiques et représentatives des problématiques actuelles, et pour une quantité moyenne à élevée de régions locales, la sélection aléatoire de régions locales se comporte aussi bien, et parfois mieux, que la sélection classique à l'aide de détecteurs de points d'intêret. Les détecteurs de points d'intêret donnent des résultats satisfaisant quand il s'agit de travailler avec un faible nombre de régions locales, mais nous montrons que le facteur le plus déterminant pour une bonne performance de classification est la quantité de régions locales observées, et les détecteurs de points d'intêret ne peuvent tout simplement pas produire suffisamment de régions pour être compétitives. Nous étudions aussi l'influence d'autres composants de la description par sac-de-mots, en particulier la taille du vocabulaire visuel, les algorithmes de création de vocabulaire visuel, la normalisation des histogrammes de distribution de régions locales et l'échelle minimum d'extraction des régions locales.

2.2 Introduction

Le chapitre précédent a présenté la reconnaissance visuelle en général, et l'a définie comme une procédure qui prend en entrée une image et qui donne en sortie la catégorie (ou le modèle) de l'objet contenu dans cette image. Nous avons aussi mentionné les applications de la reconnaissance visuelle, le principe algorithmique et l'état de l'art. Dans ce chapitre, nous nous intéressons particulièrement à l'un des algorithmes de l'état de l'art: la représentation d'images par sac-de-mots. La philosophie de cette méthode et son positionnement par rapport aux autres méthodes sont rappelés et détaillés ci-dessous.

Nous avons vu que la prédiction de la catégorie des objets contenus dans une image est un problème difficile, car l'apparence des objets au sein d'une catégorie varie grandement, à cause de modifications de position, orientation et échelle, de modifications d'illumination, d'occultations, et de la grande variabilité de formes au sein de cette classe (pensons aux différents modèles de voiture par exemple). Cette variabilité est illustrée figure 2.2. Idéalement, la représentation des images devrait-être suffisamment flexible pour couvrir un large éventail de catégories, chacune d'elles pouvant présenter de grandes variations d'apparence intra-classe, sans pour autant perdre son pouvoir discriminant d'une classe à l'autre. En d'autres termes, la généralisation intra-classe ne doit pas se faire aux dépends de la discrimination inter-classes. Les grandes variations d'apparence intra-classe et les occultations rendent impossible l'utilisation de méthodes globales telles que les méthodes à base de templates rigides ou de leurs variantes. Par contre, les approches plus locales, qui caractérisent la distribution de régions sélectionnées (ou échantillonnées) et décrites indépendament résistent bien aux difficultés portées par la variation intra-classe et les occultations. Malgré leur simplicité et l'absence de notions géométriques fines, ces méthodes permettent une description à la fois généralisée et discriminante, et elles ont fait leurs preuves dans la classification d'un grand nombre de classes ([16, 27, 50] entre autres).

Nos travaux sont basés sur la représentation d'images par sac-de-mots. L'idée générale consiste à sélectionner un ensemble de régions locales dans une image (sélection dense, aléatoire, à base de détecteurs de points d'intêret, voir figure 2.1) puis à décrire chacune d'elles à l'aide d'un descripteur visuel (descripteur SIFT, niveaux de gris normalisés). Ces descripteurs visuels sont alors quantifiés, c'est à dire ramenés à des valeurs caractéristiques, par exemple en affectant chaque descripteur à un élément d'un vocabulaire visuel précalculé (ensemble des descripteurs caractéristiques précalculés). Cela permet d'obtenir un histogramme qui comptabilise les occurrences des mots visuels (éléments du vocabulaire visuel) dans une image. Cet histogramme forme un descripteur global de l'image, utilisé pour determinier la présence ou l'absence d'un objet dans cette image, et un classifieur SVM peut être utilisé à cet effet. Pour faire de la reconnaissance visuelle avec un descripteur sac-de-mots, il faut donc déterminer comment effectuer les quatre tâches suivantes de manière optimale. Comment sélectionner les régions locales? Comment les décrire? Comment caractériser leur distribution par un vecteur de description global? Comment utiliser ce descripteur global pour déterminer la catégorie d'une image?

Notre interet principal est l'étude des différentes stratégies de sélection de régions locales, et de leur influence sur la performance de classification. La méthode de sélection est un outil critique de tout algorithme de classification par sac-de-mots. Idéalement, seules les régions les plus informatives pour la classification devraient être sélectionnées. Les détecteurs de points d'intêret multi-échelles (comme le Laplacien de Gaussiennes, Förstner, Harris affine) sont un moyen très populaire de sélectionner des régions locales [2, 16, 27, 26, 39, 49, 56, 59, 83, 91]. Cependant, bien que ces detecteurs aient prouvé leur efficacité pour des problèmes de matching, ils n'ont pas été étudiés pour sélectionner les régions les plus informatives pour la classification, et certains indices montrent qu'en effet ils ne le font pas [45, 94]. Nous montrons même que des régions sélectionnées aléatoirement sont souvent plus informatives que des régions issues de détecteurs de points d'intêret, en particulier quand les régions aléatoires sont sélectionnées en grande quantité, ce qui permet d'obtenir les meilleurs résultats de classification (voir figure 2.1). En plus de la question de la sélection de parties locales (section 2.6), nous nous intéressons aussi à d'autres étapes de la classification par sac-de-mots, à savoir la taille du vocabulaire visuel (section 2.7.1), les algorithmes de création de vocabulaire visuel (section 2.7.2), la normalisation des histogrammes de distribution de régions locales (section 2.8) et l'échelle minimum d'extraction des régions locales (section 2.9).

2.3 Travaux apparentés

La classification d'images et la reconnaissance d'objets sont des sujets très étudiés, traités par des méthodes allant du simple vote de régions locales aux complexes alignements de modèles géométriques. Dans cette section, nous ne mentionnons que les approches apparentées à la notre, c'est à dire basées sur des caractéristiques *locales* calculées sur les images. Certaines approches ont déjà été évoquées dans l'état de l'art section 1.3. Nous les classons en deux catégories, selon qu'elles utilisent des informations géométriques ou non.

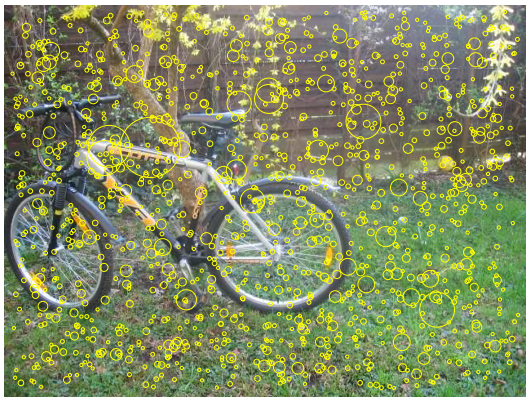
Les approches géométriques représentent un objet comme un ensemble de régions locales dont les positions sont contraintes par le modèle. Les relations entres régions locales peuvent être décrites deux à deux [2], par une constellation flexible ou une hiérarchie [27, 9], par cooccurrence [1] ou à l'aide d'un modèle géométrique rigide [56, 49].

Ces modèles géométriques sont potentiellement très puissants, mais ils sont d'une grande complexité algorithmique et souvent peu robustes aux régions locales manquantes (occultées ou non reconnues). Récemment, des modèles sac-de-mots dépourvus de contraintes géométriques sont devenus très populaires, en raison de leur simplicité, de leur robustesse, et de leurs bonnes performances en pratique. Ces modèles sont une évolution logique de l'application de méthodes d'analyse de textures par textons à la reconnaissance d'objets. L'expression "sac-de-mots" vient de l'analyse de documents [43]: les descripteurs de régions locales sont les équivalents visuels des "mots", et l'image est considérée comme un ensemble désordonné ("sac") de ceux-ci.

(1) Harris Laplace seuil élevé

(2) Harris Laplace seuil minimum

(3) Laplacien de Gaussiennes



(4) Sélection aléatoire

Figure 2.1: Exemples de méthodes de sélection multi-échelles. (1) Harris-Laplace avec un seuil de détection élevé: 465 détections. (2) Harris-Laplace avec le seuil de détection minimum: 878 détections – Noter que la sélection est toujours peu dense. (3) Laplacien de Gaussiennes: 1253 détections. (4) Sélection aléatoire: 1253 détections (quantité demandée, une infinité de détections peut être produite).

Leung et al [50] sélectionner les régions locales dans les images de manière dense, en chaque région locale ils évaluent une banque de filtres de type Gabor, et encodent la sortie de ces filtres avec un vocabulaire issu de quantification vectorielle, et composé de ce qu'ils appellent textons. Des histogrammes d'occurrences de textons sont utilisés pour reconnaitre les textures. Les textons sont aussi utilisés dans la recherche d'image par contenu [67]. Lazebnik et al [48] utilisent une sélection moins dense, puisqu'ils utilisent un detecteur de points d'intêret de type Harris affine [59]. Csurka et al. [16] utilisent l'algorithme des k-moyennes pour calculer un vocabulaire visuel à partir de régions locales sélectionnées avec un détecteur de type Harris affine et décrites avec SIFT [56]. Winn et al [94] optimisent un vocabulaire visuel calculé par l'algorithme des k-moyennes en fusionnant des cellules des histogrammes produits avec ce vocabulaire visuel. Fergus et al. [26] ont montré que les

méthodes sans géométrie permettent tout de même de localiser des objets dans des images.

Les travaux ci-dessus utilisent différentes stratégies de sélection de région locales, de description de région locales, d'encodage et quantification des régions locales, de classification globale de l'image. Les régions locales peuvent être détectées à l'aide de détecteurs de points d'intêret [2, 16, 27, 26, 39, 49, 56, 59, 83, 91], ou de manière dense, selon une grille [50, 94, 1]. Les descripteurs SIFT [16, 39, 56, 83], banque de filtres [50, 94] ou niveaux de gris [2, 27, 26, 49, 91] sont répendus. L'algorithme des k-moyennes [16, 50, 91, 94] ou un clustering agglomératif [2, 49] peuvent être utilisés pour produire un vocabulaire visuel, et il existe de nombreuses approches pour normaliser les histogrammes. Notre but est de mesurer l'influence de certains de ces choix algorithmiques sur la performance en catégorisation d'images. Notre principal intérêt n'est donc pas la rapidité, mais la performance de classification.

2.4 Bases d'images

Nous avons mené des expérimentations sur six bases d'images publiques représentatives et largement utilisées par la communauté, trois bases d'images d'objets et trois bases d'images de textures.

2.4.1 Bases d'images d'objets

Graz01 contient 667 images de taille 640×480, qui contiennent des vélos, des voitures ou des humains dans des proportions à peu près similaires (voir figure 2.2).

Xerox7[1] contient 1776 images, chacune appartenant à l'une des sept catégories suivantes: vélo, livre, bâtiment, voiture, visage, téléphone, arbre. Les catégories sont déséquilibrées, les quantités allant de 125 à 792 images par classe, et les tailles d'images sont variées (largeur allant de 51 à 2048 pixels). Des images typiques sont présentées figure 4.9.

Pascal-01[2] contient quatre catégories: voitures, vélos, motos, humains. Il y a un ensemble d'apprentissage de 684 images, et un ensemble de test de 689 images bien définis. Des images représentatives sont illustrées figure 2.3.

2.4.2 Bases d'images de textures

KTH-TIPS[3] contient 810 images de taille 200x200, avec 81 images pour les 10 catégories suivantes: papier d'aluminium, pain brun, velours côtelé, coton, biscuit, toile, peau d'orange, papier sablé, éponge et mousse. 9 textures sont illustrées figure 2.4.

[1] ftp://ftp.xrce.xerox.com/pub/ftp-ipc/

[2] http://www.pascal-network.org/challenges/VOC/

[3] http://www.nada.kth.se/cvap/databases/kth-tips/index.html

UIUCTex[4] contient 40 images par classe pour 25 catégories, mettant en oeuvre des modifications de point de vue importantes, et même des déformations non rigides. 4 textures sont illustrées figure 2.5.

Brodatz[5] contient 112 images de texture, une par classe. Il n'y a pas de modification de point de vue ou de distortions. Les images sont découpées selon une grille de taille 3x3 afin d'obtenir 9 images par classe. 25 textures sont illustrées figure 2.6.

2.5 Paramètres expérimentaux

Cette section décrit les paramètres expérimentaux utilisés tout au long des différentes études paramétriques, quand des valeurs différentes des valeurs par défaut seront utilisées nous le mentionnerons explicitement. Les detecteurs de points d'intêret multi-echelles Harris et Laplacien de Gaussiennes (LoG), ainsi que les positions sélectionnées aléatoirement, sont calculés à l'aide de la bibliothèque LAVA[6] mise au point par notre équipe. Nous utilisons les valeurs par défault suggérées par la bibliothèque, hormis le seuil de detection qui est mis

[4]http://www-cvr.ai.uiuc.edu/ponce_grp

[5]http://www.cipr.rpi.edu/resource/stills/brodatz.html

[6]http://lear.inrialpes.fr/software

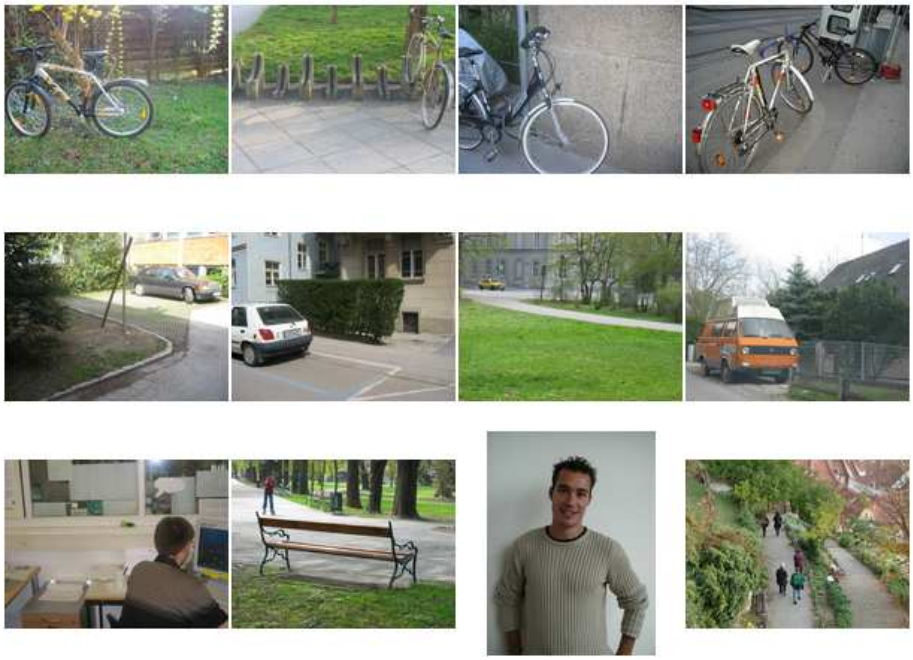


Figure 2.2: Exemple d'objets issus de la base Graz01: quatre images des catégories vélo, voiture et personne

Figure 2.3: Exemple d'objets issus de la base Pascal VOC 05: deux images des catégories vélo, voiture, motos et personnes

Figure 2.4: Neuf textures issues de la base KTH-Tips.

à 0 (pour obtenir le plus de régions possibles) et – dans le but de comparer nos travaux avec la communauté – l'échelle minimale choisie est 2, afin de supprimer les régions trop petites (voir section 2.9).

Nous utilisons des descripteurs SIFT [56], calculés une fois de plus avec la bibliothèque

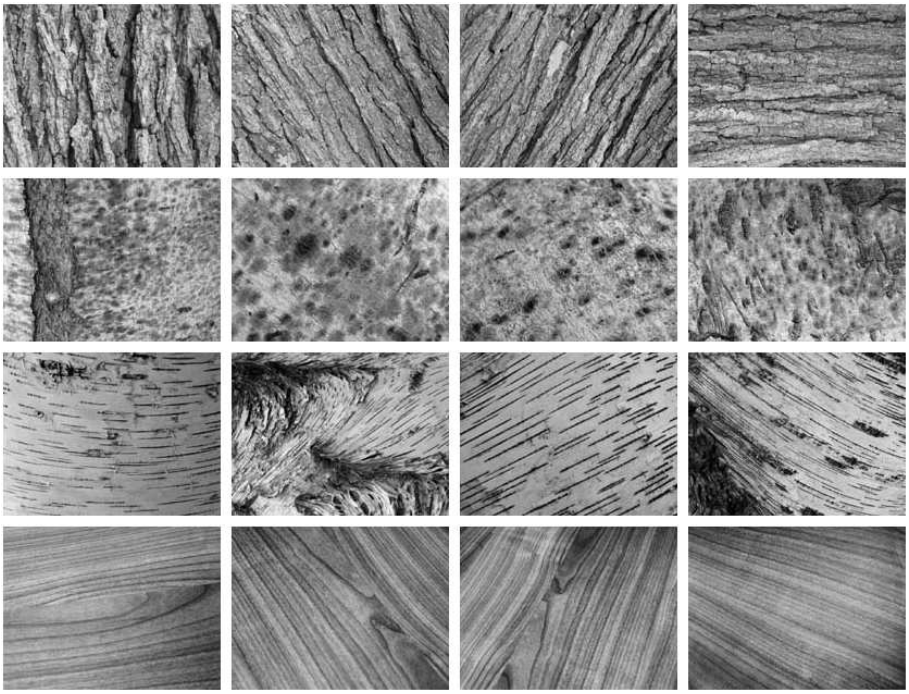


Figure 2.5: Quatre textures (une par ligne) issues de la base UIUCTex.

LAVA et des valeurs par défaut: huit orientations et une grille de quatre par quatre, ce qui produit des descripteurs de taille 128. Nous utilisons une distance euclidienne pour comparer les descripteurs SIFT, lors de la création du vocabulaire visuel ou de l'affectation d'un mot de vocabulaire à un descripteur.

Nous avons considéré un autre descripteur populaire, les niveaux de gris normalisés en luminosité (moyenne de 0) et contraste (écart-type de 1) mais comme le montre la figure 2.7, les descripteurs SIFT permettent d'atteindre une performance significativement supérieure. La supériorité des descripteurs SIFT s'explique par une plus grande invariance aux translations, à l'échelle, et par les trois étapes de normalisation: la rectification des gradients, la normalisation L2 par blocs, et le seuillage suivi d'une renormalisation L2.

Le vocabulaire visuel est calculé en initialisant ses éléments avec des descripteurs choisis aléatoirement parmi les descripteurs des régions sélectionnées dans les images. Il évolue ensuite en intégrant tous les descripteurs de toutes les régions sélectionnées dans les images d'apprentissage, grâce à un algorithme de k-moyennes en ligne. Nous utilisons une version en ligne car la version standard de l'algorithme des k-moyennes ne permettrait pas de mettre en mémoire les informations nécessaires. L'algorithme des k-moyennes en ligne est le suivant: k régions locales sont sélectionnées dans les différentes images, et définissent les mots visuels d'origine. Puis, tant qu'une condition d'arrêt n'est pas satisfaite ($10k$ itérations dans notre cas), une région locale x est sélectionnée, elle est comparée à tous les mots visuels, et

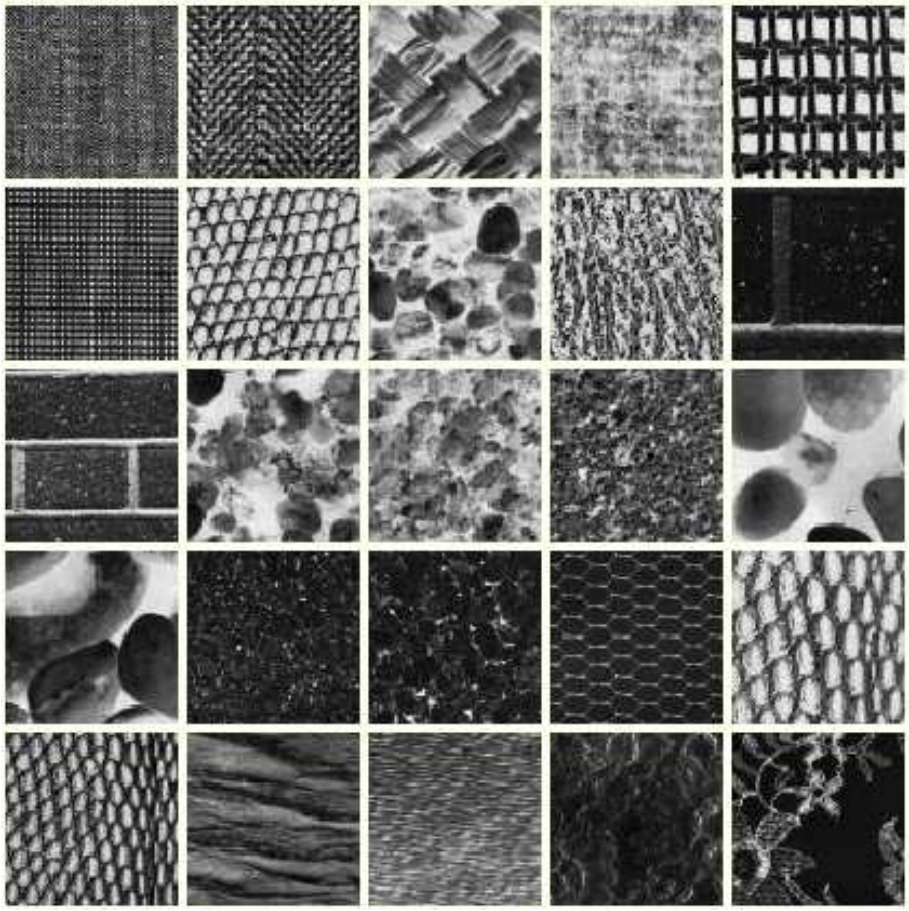


Figure 2.6: 25 textures issues de la base Brodatz.

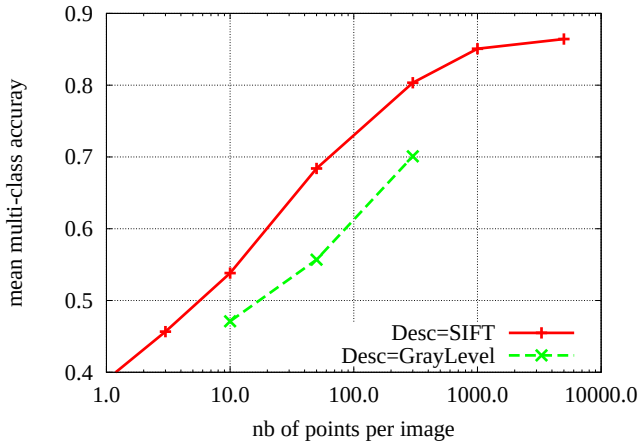


Figure 2.7: Les descripteurs SIFT surpassent largement les descripteurs par niveau de gris normalisés, ici avec une sélection aléatoire sur la base Graz.

le mot visuel w le plus similaire (distance L2) est mis à jour:

$$w \leftarrow \frac{n+1}{n+2}w + \frac{1}{n+2}x \tag{2.1}$$

où n est le nombre de mises à jour que ce mot visuel avait subies auparavant. Quand un mot visuel w est modifié pour la première fois ($n = 0$), la modification est très importante: il est transformé à moitié par la région sélectionnée x. Quand n devient grand, c'est à dire une fois qu'un mot visuel w a souvent été mis à jour, il est peu changé (à hauteur de $1/(n+2)$) par les régions sélectionnées.

Le descripteur global d'une image est calculé en assignant à chaque région locale sélectionnée le mot visuel (issu du vocabulaire visuel précédemment calculé) le plus proche (distance L2). Cela produit un histogramme d'occurrences de chaque mot du vocabulaire visuel. Cet histogramme est un vecteur descripteur qui a la meme taille que le vocabulaire visuel. Nous considérons trois méthodes de normalisation de cet histogramme (détails section 2.8): le comptage brut; la binarisation simple (une ou plusieurs apparitions sont considérées comme identiques, et ne comptent que pour un); et une binarisation adaptative que nous proposons, qui choisit le seuil de binarisation dimension par dimension en optimisant un critère d'information mutuelle (MI). Le comptage brut donne des performances bien en dessous des deux autres méthodes, nous ne présentons donc pas les résultats avec cette méthode de normalisation.

Un classifieur SVM *un contre un* est utilisé pour la classification, l'implémentation utilisée est LibSVM[7]. Pour la classification multiclasses, la classe prédite est celle qui recueille le plus de votes parmi toutes les compétitions de classes deux à deux. Les classifieurs ont des noyaux linéaires, hormis dans la section 2.10 ou nous utilisons un noyau Gaussien pour nous comparer à des travaux utilisant des noyaux non linéaires.

La mesure de performance utilisée est la moyenne non pondérée du taux de classification correcte par classe. Cela reflète mieux le comportement de l'algorithme que la moyenne pondérée par la taille des classes. En effet, si une classe est sur représentée, un classifieur qui prédirait systématiquement cette classe aurait une très bonne performance. Ce n'est pas le cas avec la moyenne non pondérée. Chaque expérimentation est reproduite six fois intégralement, de la sélection des régions locales à la mesure de performance, en passant par le calcul de vocabulaire visuel. La performance que nous reportons est la moyenne des six performances calculées. Seule la base Pascal-01 dispose d'un ensemble d'apprentissage et de test bien défini. Pour les autres bases, nous utilisons de la validation croisée en deux paquets.

2.6 Méthode de sélection de régions locales

L'utilisation d'un ensemble de régions locales indépendantes a fait ses preuves pour la catégorisation d'images et le reconnaissance d'objets. Une question reste ouverte: comment

[7] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

sélectionner ces régions? Les objets peuvent apparaitre à n'importe quelle position et à n'importe quelle échelle dans les images, donc les régions locales doivent être extraites à plusieurs échelles [50, 94]. La sélection dense de régions locales [45, 94], qui consiste à traiter tous les pixels à toutes les échelles, capture le plus d'information, mais nécessite des ressources en temps et en mémoire très importantes, la plupart du temps de calcul étant passée à traiter des régions peu informatives. Plusieurs auteurs soutiennent que les calculs peuvent être accélérés, et la performance peut être améliorée, en utilisant une mesure de saillance, afin de ne sélectionner que les régions informatives. La reconnaissance d'images à partir d'exemples consiste globalement à faire correspondre une nouvelle image aux images d'une base de données, et il est tentant d'investiguer ce type de méthode dans le cadre de la reconnaissance de catégories d'objets. De ce fait, de nombreux auteurs ont utilisé des détecteurs de points d'intêret génériques [2, 16, 27, 39, 49, 56, 59, 83, 91]. Ces méthodes de sélection de régions locales présentent l'avantage d'être répétables[56, 59] et invariantes aux modifications d'échelle, de translation, de rotation, et même dans une certaine mesures aux déformations affines [60].

Cependant, les detecteurs de points d'interet ou les mesures de saillance sont basées sur des mesures bas niveau dans les images, sans relation directe avec le pouvoir discriminant des régions locales pour la classification et la reconnaissance. Dans les travaux précédents, aucun auteur n'a vérifié que les régions locales séléctionnées étaient plus discriminantes que des régions locales choisies aléatoirement. De plus, il est évident que l'un des facteurs principaux influençant la performance de classification est la quantité de régions locales considérées par image, et cela n'a pas non plus été étudié dans les travaux précédents.

Nous nous intéressons à ces problèmes en comparant trois méthodes de sélection de régions locales. Le *Laplacien de Gaussiennes (LoG)*: un détecteur multi-échelles proposé par [51] et popularisé par [56]. Le détecteur *Harris-Laplace (Harris)*: le détecteur multi-échelles (non affine) utilisé dans [48]. Enfin, le *détecteur aléatoire (Rand)*: les régions locales sont sélectionnées aléatoirement dans une pyramide de grilles régulières, toutes les régions locales étant équiprobables, les petites échelles prédominent. Dans tous les cas, nous construisons un vocabulaire visuel de 1000 éléments par l'algorithme des k-moyennes en ligne, et nous utilisons une normalisation MI (détaillée en section 2.8) et un classifieur SVM linéaire.

La figure 2.8 montre la performance moyenne de la classification multiclasses en fonction du nombre de régions locales sélectionnées pour les différentes bases d'images. Il s'agit plus précisément de la moyenne de la performance sur six expérimentations complètes, des valeurs typiques d'écart type peuvent être lues dans le tableau 2.1. Chaque courbe montre l'effet de l'augmentation du nombre de régions locales utilisées pour la description d'une image. Pour les détecteurs de points d'intêret, la quantité de régions désirée est obtenue en faisant varier le seuil de détection. En pratique, ces detecteurs ne peuvent retourner qu'un nombre limité de points, même quand le seuil de détection est mis à zéro. Cela apparaît clairement dans les différentes courbes. C'est l'un des facteurs limitants principaux des méthodes à base de detecteurs de points d'intêret: elles ne peuvent tout simplement pas fournir une sélection de régions locales suffisemment dense pour produire les meilleurs résultats

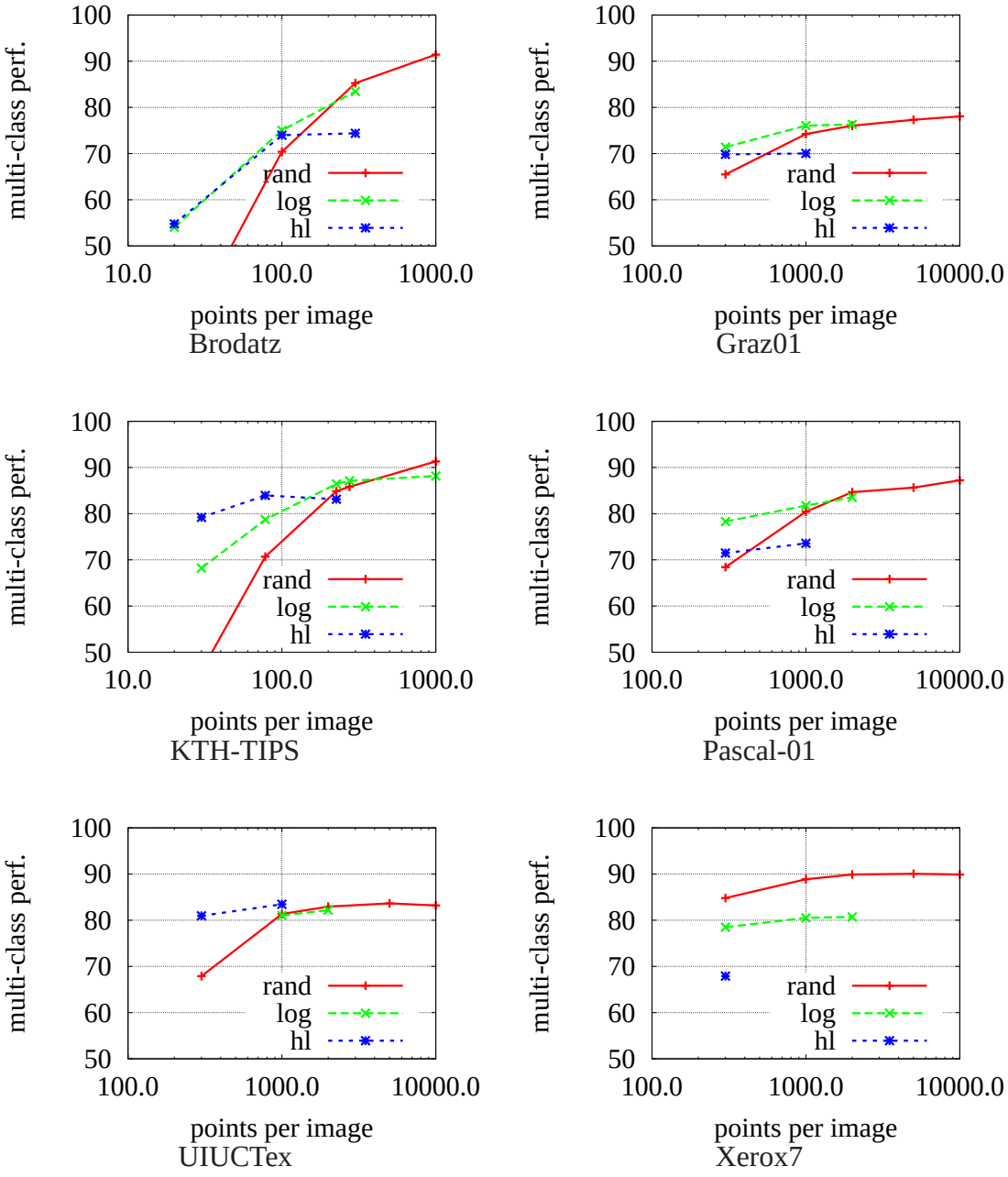


Figure 2.8: Performance moyenne de la classification multiclasses sur les six bases de données en fonction du nombre de régions locales sélectionnées.

de classification. Dans la très grande majorité des cas, la performance augmente quand la quantité de régions locales considérées augmente, et la sélection de régions locales aléatoire finit toujours par surpasser les autres, car il peut produire une quantité illimitée de régions.

Pour les régions issues de detecteurs de points d'intêret, il apparaît que les régions peu saillantes (détectées uniquement avec un seuil très bas, et donc apparaissant uniquement quand un grand nombre de régions sont incluses) permettent toujours d'améliorer la performance, ce qui encore une fois nous encourage à utiliser des régions sélectionnées aléatoirement. Cependant, la saturation de la progression apparait plus tôt pour ces régions issues de detecteurs de points d'intêret.

Dans des conditions où peu de régions locales doivent être utilisées, la sélection par détecteurs de points d'intêret se montre souvent plus performant que la sélection aléatoire mais on ne peut globalement déterminer quel détecteur surpasse l'autre. D'ailleurs, pour la base d'images Xerox7, c'est la sélection aléatoire qui donne les meilleurs résultats, même pour une faible quantité de régions aléatoires.

En général, si la performance est la seule préoccupation, alors la sélection aléatoire d'une grande quantité de régions est la meilleure option.

2.7 Vocabulaire Visuel

Cette section étudie l'influence de la quantification vectorielle utilisée pour créer le vocabulaire visuel sur les performances de classification.

2.7.1 Taille du vocabulaire visuel

La taille du vocabulaire visuel, c'est à dire le nombre de mots qu'il contient, est un élément fondamental pour une performance élevée. En effet, [16] observe une augmentation importante de la performance quand la taille du vocabulaire augmente. Nous avons reproduit des expérimenations similaires, en utilisant un algorithme de k-moyennes en ligne (plutôt que k-moyennes classique), et avons expérimenté des tailles de vocabulaires visuels encore plus élevées que dans [16]. Nous avons aussi étudié l'influence conjointe de la taille du vocabulaire visuel et du nombre de régions sélectionnées par image. Les résultats sont reportés figure 2.9, qui indique la performance moyenne sur 6 expérimentations, pour la base Xerox7, pour chacun des détecteurs. Les réglages paramétriques sont identiques à ceux des sections précédentes. Pour chaque détecteur, il y a des gains importants quand on passe d'une faible taille de vocabulaire à une taille plus raisonnable, mais on se retrouve en situtation de sur-apprentissage pour des vocabulaires très larges (4000 elements). Quand le nombre de points sélectionnés par image devient élevé, la sélection avec détecteurs de points d'intêret amène finalement à faire du sur-apprentissage, ce qui n'est pas le cas avec des points sélectionnés aléatoirement. Il semble ne pas y avoir de forte interaction entre la taille du vocabulaire visuel et le nombre de régions sélectionnées par image. Bien sûr, la taille et la diversité de l'ensemble d'apprentissage jouent un rôle important sur la saturation dans l'augmentation de la performance, et sur le sur-apprentissage. Nous n'avons pas fait d'études détaillées à ce sujet. Les observations faites sur la base Xerox7 se généralisent aux cinq autres bases considérées.

Base	Rand KM	Rand no KM	LoG KM	LoG no KM	H-L KM	H-L no KM
Graz01	74.2 ± 0.9	71.3 ± 0.9	76.1 ± 0.5	72.8 ± 0.9	70.0 ± 1.4	68.8 ± 2.0
KTHTIPS	91.3 ± 1.1	92.1 ± 0.4	88.2 ± 1.0	85.0 ± 1.8	83.1 ± 2.1	81.3 ± 1.1
Pascal-01	80.4 ± 1.4	77.4 ± 0.9	81.7 ± 1.0	78.7 ± 2.3	73.6 ± 2.3	67.8 ± 2.8
UIUCTex	81.3 ± 0.8	75.2 ± 1.4	81.0 ± 1.0	76.0 ± 0.8	83.5 ± 0.8	80.4 ± 0.8
Xerox7	88.9 ± 1.3	87.8 ± 0.5	80.5 ± 0.6	79.9 ± 0.9	66.6 ± 1.8	65.6 ± 1.5

Table 2.1: Influence des méthodes de calcul du vocabulaire visuel. Ce tableau indique la moyenne et l'écart type, calculés sur 6 expérimentations complètes, de la performance en classification multiclasses pour diverses bases de données, divers détecteurs – détecteur aléatoire Rand, Laplacien de Gaussiennes LoG, Harris-Laplace H-L – et des vocabulaires visuels appris avec l'algorithme des k-moyennes en ligne (KM) ou des vocabulaires visuels faits de mots visuels sélectionnés aléatoirement (no KM).

2.7.2 Algorithme de construction du vocabulaire visuel

La section 2.5 présente deux méthodes de construction de vocabulaire visuel: choisir des mots aléatoirement parmi les descripteurs d'une base d'exemples, ou un algorithme de k-moyennes en ligne. Notons qu'un vocabulaire visuel calculé par l'algorithme des k-moyennes en ligne est justement initialisé par un vocabulaire visuel fait de mots choisis aléatoirement, puis que ce vocabulaire visuel est modifié pour mieux représenter la distribution des mots de la base d'apprentissage. Le tableau 2.1 compare ces deux méthodes, une fois de plus avec un vocabulaire de 1000 éléments, une normalisation des histogrammes par information mutuelle et un classifieur SVM linéaire. On sélectionne 1000 régions par image, ou moins si les détecteurs de points d'intérêt ne peuvent en sélectionner autant. A l'exception d'un seul cas (KTH-TIPS avec régions aléatoires), le vocabulaire optimisé par algorithme des k-moyennes en ligne est plus performant que le vocabulaire fait de mots aléatoires. Le gain moyen (2.7%) est statistiquement significatif, mais chacun des gains individuels ne l'est pas. On observe que les vocabulaires faits de mots choisis aléatoirement sont un peu moins bons que les autres, mais que leur performance est tout de même élevée. L'optimisation des mots par l'algorithme des k-moyennes en ligne apporte un gain faible (mais néanmoins utile), cependant celui-ci est moindre que celui resultant d'une augmentation de la taille du vocabulaire visuel, ou du nombre de régions sélectionnées.

2.7.3 Images utilisées pour la construction du vocabulaire visuel

Etant donnée la performance du vocabulaire fait de mots choisis aléatoirement dans la base d'apprentissage (voir section précédente), on peut aussi se demander s'il est nécessaire de construire un vocabulaire visuel spécifique par base d'images, ou bien si un vocabulaire visuel générique pourrait suffire (voir aussi [94]).

La figure 2.10 donne le taux d'erreur de classification pour trois vocabulaires visuels, évalués sur les bases KTH-Tips (texture) et Graz (catégories d'objets). Les vocabulaires considérés sont (a) vocabulaire appris sur la base KTH-Tips par algorithme des k-moyennes (b) vocabulaire appris sur la base Graz par algorithme des k-moyennes (c) vocabulaire totalement aléatoire. Ce dernier est constitué d'un ensemble de vecteurs de 128 dimensions, fait de valeurs aléatoires positives, et normalisées comme les descripteurs SIFT pour que la somme des carrés fasse un.

Comme on s'y attend, le vocabulaire visuel appris sur la base KTH donne les meilleures performances sur la base KTH, et le vocabulaire visuel Graz donne les meilleures performances sur la base Graz. Le vocabulaire totalement aléatoire obtient des résultats significativement inférieurs aux vocabulaires issus d'images réelles, mais la performance reste bien meilleure que le hasard.

On en conclut donc que même un encodage complètement aléatoire est capable d'encoder des informations discriminantes, mais qu'une quantification issue de données réelles augmente très significativement la performance.

2.8 Normalisation des histogrammes

La description de toutes les images permet d'obtenir une matrice de comptage des occurrences des mots visuels, analogue à la matrice document-mot en analyse de texte. Les colonnes font référence aux éléments du vocabulaire visuel, et chaque ligne est un histogramme non normalisé qui compte les occurrences des différents mots visuels sélectionnés dans une image. Ainsi, l'élément situé ligne i et colonne j indique le nombre de fois que le mot visuel numéro j a été sélectionné dans l'image i.

Comme pour l'analyse de texte, l'utilisation directe de ce comptage dans un classifieur n'est pas optimale, en particulier pour les classifieurs SVM linéaires (voir [68]), en raison de la grande différence d'occurrence moyenne des différents mots. Différentes méthodes de normalisation ont été étudiées. Nous comparons ici deux méthodes, qui consistent à effectuer des normalisations par ligne qui binarisent les histogrammes. La première méthode est une simple binarisation: pour chaque image, tout mot visuel sélectionné met l'élément correspondant dans la matrice à 1, sinon la valeur affectée est 0. La seconde méthode choisit le seuil de binarisation qui maximise l'information mutuelle entre le mot visuel et les différentes catégories [68] (détails section 4.6.2). Comme précédemment, nous utilisons un vocabulaire visuel de 1000 éléments, l'algorithme des k-moyennes en ligne, et un SVM linéaire. Les résultats sur deux bases de données sont présentés figure 2.11 – les autres bases de données donnent des résultats similaires.

Aucune méthode ne prédomine dans toutes les situations, mais le seuillage adaptatif est clairement préférable quand les détecteurs retournent une grande quantité de régions (ici 10000 points pour 1000 mots visuels). Par exemple, sur la base Xerox7, avec 1000 régions locales par image et 1000 centres, la densité de l'histogramme est de 27%, cette densité atteint 43% avec 10000 régions locales par image. Le seuillage adaptatif permet de réduire

la densité à 13% dans le dernier cas, ce qui permet au SVM de se concentrer sur les données les plus significatives.

2.9 Echelle minimum de sélection des régions locales

Idéalement le classifieur devrait exploiter des informations provenant de toutes les échelles auxquelles les objets sont observés. Pour ce faire, il faudrait que la sélection et la description des régions locales se fassent avec une bonne invariance aux changements d'échelles, car les objets (et donc leurs parties caracteristiques) peuvent avoir des tailles quelconques dans les images Il faudrait aussi un classifieur qui puisse tirer parti de détails fins. Cela requiert au minimum de disposer d'un vocabulaire visuel assez riche pour encoder les informations fines et plus grossières sans les mélanger. La qualité de l'extraction de régions locales dans les petites échelles (i.e. produisant de petites régions locales dans les images d'origine) est alors cruciale, car les petites échelles contiennent la plupart des régions discriminantes, ainsi que du bruit.

En pratique, on utilise une échelle minimale pour l'extraction des régions d'intêret. Cette section évalue l'influence de cette échelle minimum sur la performance en classification. On utilise comme précédemment un vocabulaire visuel de 1000 éléments, calculé par algorithme en ligne des k-moyennes, une normalisation MI et un classifieur SVM linéaire.

La figure 2.12 montre l'évolution de la performance moyenne sur les bases Brodatz et Xerox7 quand l'échelle minimum varie entre 1 et 3. Toutes les autres expérimenations utilisaient une échelle minimum de 2. Les descripteurs SIFT de la biliothèque LAVA utlisent des blocs 4x4, et la première échelle calcule des descripteurs sur 12x12 pixels, donc chaque bloc contient au minimum 3x3 pixels. La performance des détecteurs Laplacien de Gaussiennes et Harris diminue significativement quand l'échelle minimum augmente: les détecteurs retournent de moins en moins de régions locales, et les informations locales sur les classes disparaissent. Avec une sélection aléatoire, le nombre de régions locales ne varie pas, et aucune tendance ne se dessine: sur la base Brodatz, les petites échelles peuvent gêner, sur la base Xerox7 elles sont très utiles.

On en conclut que les petites échelles servent aux détecteurs de points d'interet car s'en priver diminue drastiquement la quantité de points que peut utiliser le classifieur. Le detecteur aléatoire ne souffre pas de ce biais. Ainsi, sur certaines bases, ignorer les petites échelles (qui peuvent contenir beaucoup de bruit) peut augmenter la performance.

2.10 Resultats sur les bases du challenge Pascal 2005

Les sections précédentes ont montré l'intêret de la sélection aléatoire, et ont mesuré l'influence des divers paramètres. Ici, nous montrons qu'il suffit de sélectionner suffisemment de régions locales pour dépasser les méthodes de l'état de l'art les plus performantes. Cela est illustré sur la base du challenge Pascal 2005 VOC (Visual Object Classification), parce que plusieurs équipes ont publié des résultats sur cette base [19].

Nous utilisons les réglages suivants: 10 000 régions locales sélectionnées par image, algorithme des k-moyennes en ligne, normalisation par taux d'information mutuelle, SVM à noyau RBF dont le paramètre γ est égal au médian des distances entre les descripteurs d'apprentissage, et un vocabulaire visuel de taille 1000 ('Rand1k') ou 4000 ('Rand4k').

Les résultats sont présentés sur la figure 2.13 sous forme de courbe ROC. C'est naturellement le plus large vocabulaire visuel ('Rand4k') qui prédomine. Le tableau 2.2 compare Rand4k aux résultats obtenus pendant le challenge, ainsi qu'aux travaux de Zhang et al [98]. Durant le challenge (ligne 'Top Pascal'), chaque compétition a été remportée par un algorithme différent, alors que nos résultats utilisent une seule méthode, et des valeurs de paramètres fixes d'une compétition à l'autre. La méthode proposée par [98] utilise une combinaison de detecteurs de points d'intêret complexes (Harris-Scale et Laplacian-Scale) et un noyau de SVM spécifiquement développé (Earth Movers Distance), et nos résultats sont meilleurs alors que nous n'utilisons que des régions choisies aléatoirement (en grande quantité) et un noyauu RBF classique.

2.11 Conclusions

Nous avons mené une observation expérimentale de l'influence du choix des différents composants d'une description d'images par sac-de-mots en vue de faire de la catégorisation d'images. Pour ce faire, nous avons étudié l'influence des stragégies de sélection des régions locales, de la génération du vocabulaire, de la normalisation des histogrammes, sur un ensemble de bases de données représentatives et largement utilisées par la communauté pour la classification d'images.

Notre observation principale concerne les stratégies de sélection des régions locales. Sous la contrainte d'utiliser peu de régions locales, les détécteurs de points d'interêt tels que Harris-Laplace et Laplacien de Gaussiennes donnent des performances correctes. Mais si le but est d'obtenir la meilleure performance, alors ils ne peuvent pas retourner assez de régions locales, et sont donc moins performants qu'une simple sélection aléatoire d'une grande quantité de régions locales. Dans toutes nos expérimentations, la quantité de régions locales sélectionnées est de loin le paramètre le plus influent sur le taux de classification. Si on cherche à utiliser peu de régions locales, aucune stratégie de sélection (Harris-Laplace, Différence de Gausssiennes, détecteur aléatoire) ne domine sur toutes les bases. Par contre, quand on utilise une grande quantité de régions locales, on obtient la meilleure performance, et c'est le détecteur aléatoire qui prédomine à chaque fois. La supériorité du détecteur aléatoire est due au fait ques les détecteurs de points d'intêret retournent des maxima locaux, et donc qu'ils ne peuvent retourner qu'un nombre assez limité de régions locales. Les mesures de saillance optimisées par les détecteurs de points d'intêret ont fait leurs preuves dans des applications de matching, mais nous avons montré que ce n'est pas le cas dans les applications de catégorisation d'images. Cela nous amène à nous questionner sur des résultats comparatifs publiés dans la littérature, car ils ne prennent pas en compte le nombre de régions locales sélectionnées, et nous avons montré qu'une simple sélection aléatoire d'un

Méthode	motos	vélos	persones	voiture	moyenne
La notre (rand4k)	97.6	93.8	94.0	96.1	95.4
Top Pascal [19]	97.7	93.0	91.7	96.1	94.6
Zhang et al [98]	96.2	90.3	91.6	93.0	92.8

Table 2.2: Comparaison entre notre méthode Rand4k, les meilleurs résultats obtenus pendant le challenge (par différentes méthodes), et la méthode à base de détecteur de points d'intêret de Zhang et al. Les chiffres représentent le taux d'erreur égale des courbes ROC.

grand nombre de régions permet de surpasser les méthodes d'apprentissage les plus sophistiquées (section 2.10).

De même, pour les méthodes multi-échelles, l'échelle minimum à laquelle les régions locales sont sélectionnées a une influence considérable sur les performances, car la vaste majorité des régions locales utiles à la classification apparaissent aux échelles les plus fines, et rentrent en concurrence avec le bruit. En raison de ce compromis information/bruit des petites échelles, il peut être utile de conserver ou non ces petites échelles, en fonction des bases de données. Cela remet en question l'invariance d'échelle annoncée par les méthodes de représentation d'images par sacs de mots.

Enfin, il est vrai que la performance augmente avec la taille du vocabulaire visuel, mais celle-ci sature assez rapidement. Bien que les méthodes de création de vocabulaire aient une influence sur la performance, nous avons vu qu'un vocabulaire aléatoire donne des performances très honorables, ce qui montre qu'il ne faut pas s'attendre à de grandes améliorations de performances en améliorant les méthodes de construction du vocabulaire visuel.

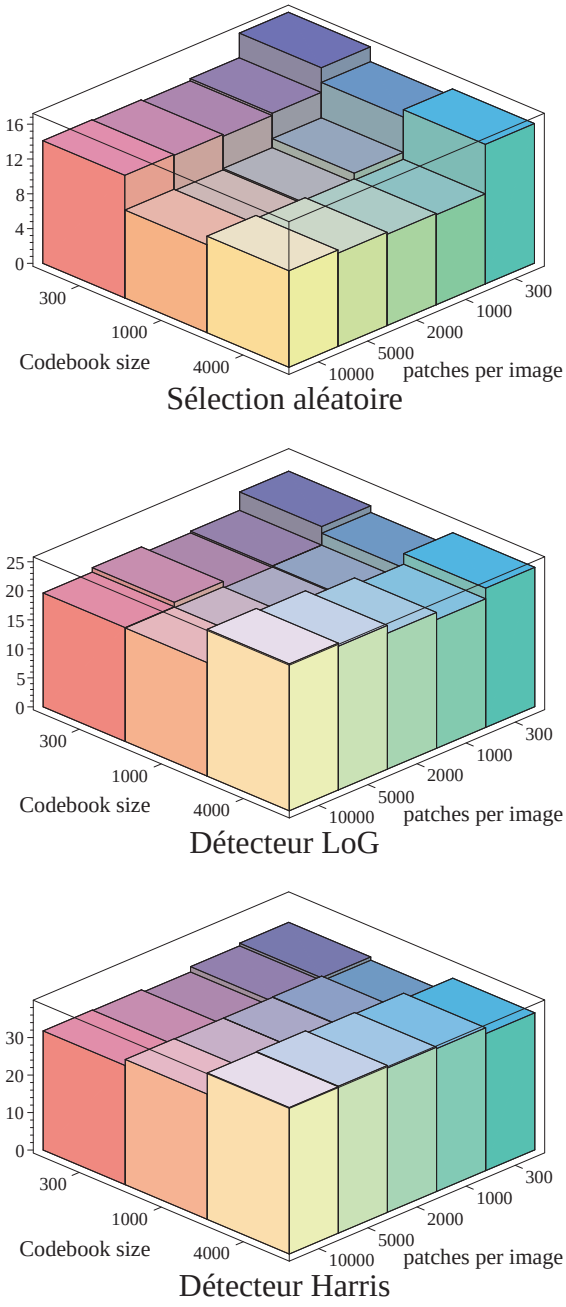


Figure 2.9: Influence de la taille du vocabulaire visuel et du nombre de régions sélectionnées par image pour diverses stratégies de sélection. Les valeurs reportées sont les moyennes du taux d'erreur de classification.

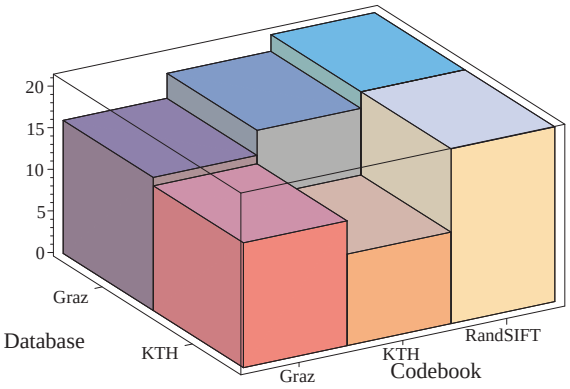


Figure 2.10: Influence du choix des images utilisées pour apprendre le vocabulaire visuel, pour un vocabulaire appris sur KTH, Graz, ou fait de vecteurs SIFTs aléatoires, évalués sur les bases KTH et Graz. Les valeurs reportées sont les moyennes du taux d'erreur de classification.

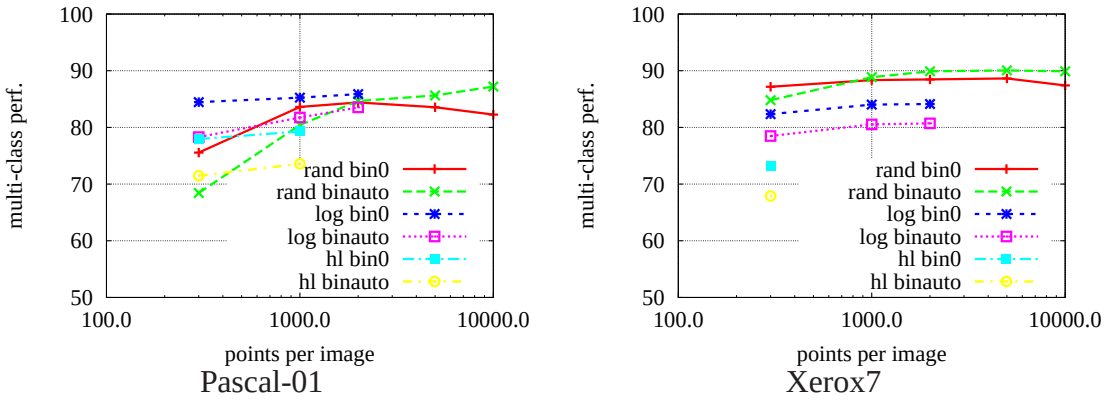


Figure 2.11: Influence de la méthode de normalisation des histogrammes sur le taux de classification moyen, sur les bases Pascal-01 et Xerox7. Les histogrammes sont binarisés avec un test nul/non nul (bin0) ou en choisissant les seuils qui maximisent le taux d'information mutuelle entre le mot de vocabulaire visuel et les catégories (binauto). Le seuillage adaptatif est préférable pour les détecteurs aléatoires, car certaines cellules de l'histogramme contiennent de grandes quantités de votes.

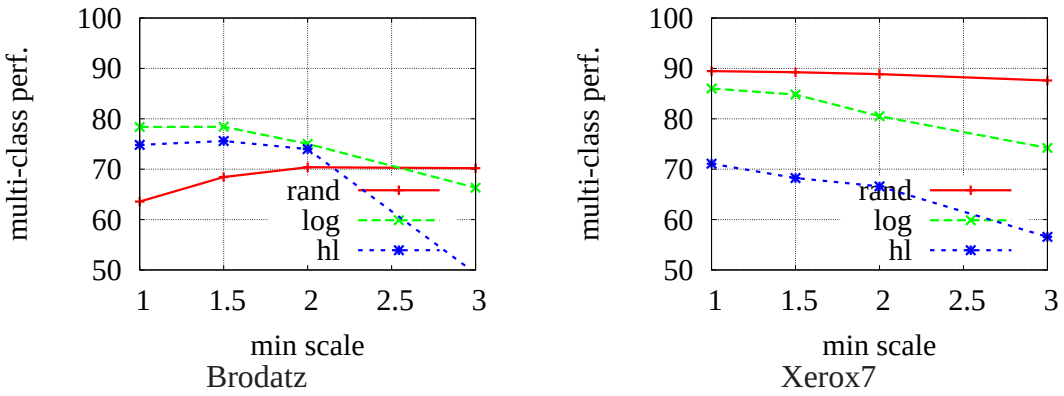


Figure 2.12: Influence de l'échelle minimum des régions locales décrites par SIFT sur la base Brodatz et Xerox7. Performance multiclasses en fonction de l'échelle minimum selectionnée.

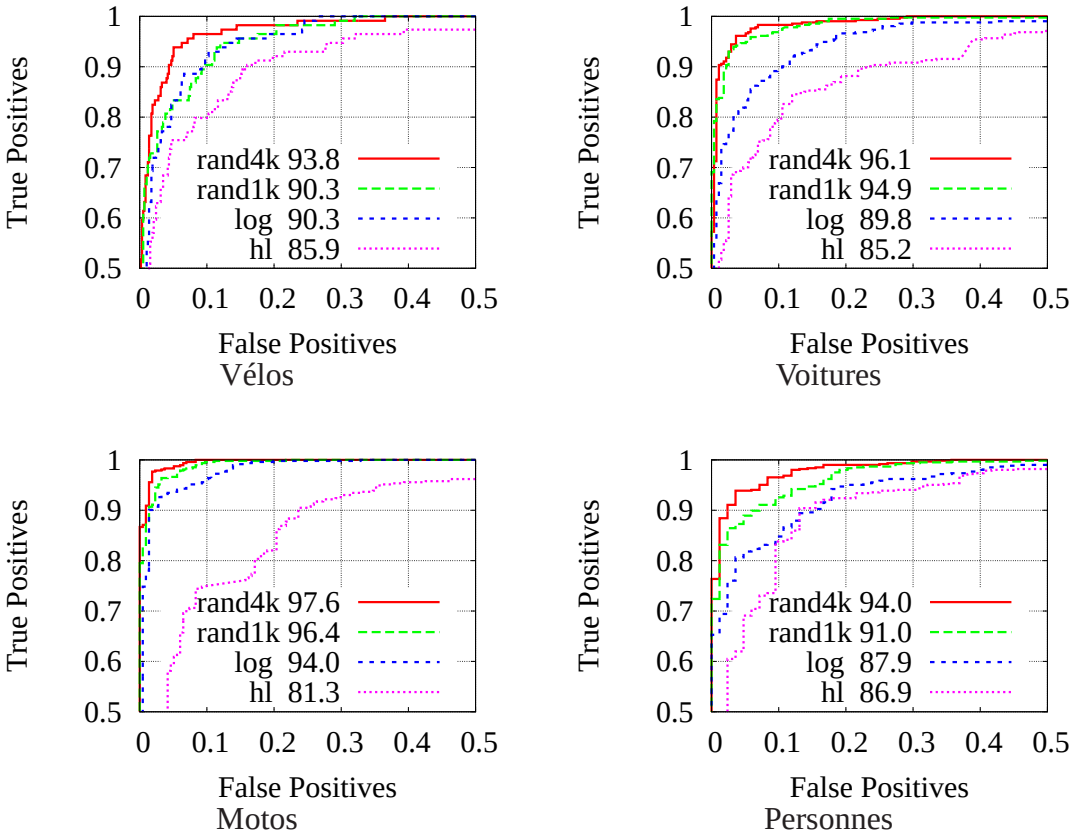


Figure 2.13: Courbes ROC pour les 4 categories du challenge PASCAL 2005 VOC: vélos, voitures, motos, personnes. Les vocabulaires visuels ont 1000 éléments, sauf pour rand4k qui en a 4000. Nous indiquons à chaque fois l'EER (equal error rate). Les régions locales sont sélectionnées par Laplaciens de Gaussiennes (log), Harris-Laplace (hl), selection aléatoire de 1000 (rand1k) ou 4000 (rand4k) régions.

THIRD CHAPTER

Descriptions sacs-de-mots d'images infra-rouge

3.1 Résumé du chapitre

La description d'images par méthode sac-de-mots est maintenant une méthode de référence de l'état de l'art. Nous avons mené une étude algorithmique approfondie de cette méthode pour des images des bases standard de l'état de l'art (voir chapitre 2). Le présent chapitre étudie les performances des méthodes sacs-de-mots en imagerie infra-rouge. L'infra-rouge présente des spécificités (bruit, faible résolution, variabilité intra-classe,...) et il convient de déterminer si la description sac-de-mots se comporte aussi bien en imagerie infra-rouge qu'en imagerie visible. Nous menons deux études: une étude algorithmique, qui ne considère que les paramètres intervenant dans l'algorithme (comme la taille du vocabulaire visuel) et une étude opérationnelle, qui ne considère que les paramètres intervenant sur le terrain (comme la distance véhicule-caméra). Les conclusions principales sont que (a) les paramètres algorithmiques se comportent comme ceux présentés en imagerie visible dans le chapitre 2 (b) les paramètres opérationnels les plus sensibles sont la distance et le taux d'occultation.

3.2 Introduction

Dans le chapitre précédent, nous avons étudié en détails l'intérêt de la représentation d'images par sac-de-mots pour la reconnaissance visuelle. Dans ce chapitre, nous étudions en particulier l'intérêt de ces méthodes pour la description des images infra-rouges qui intéressent la société Bertin Technologies. En effet, les travaux de thèse ont été effectués dans le cadre d'une thèse CIFRE, avec des applications industrielles concrètes. L'une d'entre elles est la reconnaissance de véhicules filmés par une caméra infra-rouge.

Le contexte général dans lequel s'inscrit la reconnaissance d'objets militaires en vidéo infra-rouge se nomme DRI: Detection, Reconnaissance, Identification. Ces notions ont été évoquées dans l'introduction (chapitre 1) et nous les détaillons ci-dessous. La *détection* consiste à déterminer la région de l'image où se situe une cible en mouvement. Elle s'effectue en général par une analyse de mouvement. La technique la plus répendue est la soustraction de fond quand la caméra est immobile: une image de fond de scène est calculée, et celle-ci est soustraite à l'image courante acquise par la caméra: les zones où les différences sont importantes contiennent potentiellement un objet qui ne fait pas parti du fond, et donc une cible. La *reconnaissance* consiste à déterminer la catégorie grossière de la cible: humain, véhicule léger, véhicule lourd, animal. La reconnaissance peut s'effectuer à l'aide de techniques d'analyse cinématiques: des humains, des véhicules, des animaux ont des vitesses et des trajectoires typiques différentes. Elle peut aussi s'effectuer par des analyses de formes grossières: ratios hauteur largeur, position des points chauds, etc. *L'identification* consiste à déterminer le type précis d'objet: le modèle de véhicule, l'espece de l'animal, le type ami/ennemi de l'humain, etc. L'identification se fait en général à l'aide de matching de formes caractéristiques. Par exemple, les apparences d'un tank sont stockées pour plusieurs orientations, plusieurs distances, plusieurs conditions météorologiques, plusieurs conditions thermiques, et lorsque qu'on cherche à identifier une région d'intéret fournie par la partie Détection, on compare toutes les images de tank à la région fournie. Si la distance entre une image de référence et la région d'intérêt est inférieure à un seuil, alors on prédit que la région d'intérêt contient un tank. Un autre algorithme consiste à détecter les points chauds dans la région d'intérêt, à effectuer un seuillage puis à calculer les contours de l'objet, un score est alors calculé entre la silhouette extraite et les silhouettes de référence stockées dans une base de données. En fonction du score obtenu, le véhicule est identifié.

Ce chapitre s'intéresse aux phases de reconnaissance et d'identification. Comme nous pouvons le constater dans le paragraphe précédent, les méthodes utilisées en général sont des méthodes globales, et comme nous l'avons indiqué dans l'état de l'art section 1.3, ce type de méthode n'est pas robustes à des variations non uniformes de l'image, ce qui est très fréquent dans des conditions réelles. Nous proposons donc d'utiliser des méthodes locales pour la reconnaissance et l'identification, et nous traitons les deux problèmes simultanément. Nous allons donc appliquer la méthode de description d'images par sac-de-mots aux problèmes de reconnaissance et d'identification en imagerie infra-rouge.

L'infra-rouge a des spécificités par rapport à l'imagerie visible. Il convient d'analyser si les méthodes proposées au chapitre 2 sont applicables à l'imagerie infra-rouge, malgré

ses spécificités. Des exemples typiques d'images infra-rouge sont visibles figure 3.18. Les images qui nous intéressent sont caractérisées par une faible résolution spatiale: les objets sont de petite taille, de XX à XX pixels de hauteur. Elles sont en général très bruitées, en raison du gain appliqué sur le capteur pour obtenir une quantité minimum de signal. Ce gain amplifie aussi le bruit. Le signal reçu par la caméra est fortement atténué, il diminue avec le carré de la distance et l'exponentielle du taux d'humidité. De plus, il est fréquent de constater que le contraste interne d'un véhicule, et le contraste véhicule/fond, sont faibles, ce qui rend la reconnaissance difficile. Enfin, la plus grande spécificité est la grande variabilité intra-classe. Les véhicules sont des sources de rayonnement infra-rouge, et à chaque étape de l'augmentation de température des différentes parties (moteur, roues, carrosserie) l'apparence varie. Le but de cette section est d'évaluer les algorithmes proposés au chapitre 2: quels sont les paramètres algorithmiques (tels que la taille du vocabulaire visuel) sensibles? quels sont les paramètres opérationnels (tels que la distance véhicule-caméra) sensibles?

Ce type d'études intéresse surtout les militaires, et les résultats sont souvent confidentiels. Pour cette raison, nous n'avons pas trouvé d'étude similaire dans la bibliographie. Une telle étude n'est pas facile à mener en raison de la difficulté à obtenir des données correctes. Il faut disposer en effet d'une grande quantité d'images infra-rouge pour mener les différentes études paramétriques. Il faut tout d'abord des données sur plusieurs catégories de véhicules, sinon il s'agit d'une tâche de détection (objet/fond). Pour l'étude algorithmique, il faut disposer d'une certaine quantité d'images, toutes conditions d'acquisition confondues. Pour l'étude opérationnelle, il faut faire varier les paramètres que l'on cherche à étudier. Si l'on s'intéresse à l'influence de la distance et de l'atténuation atmosphérique, alors il faut des données où ces paramètres varient. Cela peut vite nécessiter une grande quantité de données. Ce problème a été résolu par le CEP Arceuil (Organisme dépendant de la Direction Générale de l'Armement avec qui Bertin Technologies est en relation) par la production d'images hybrides [47]: les images sont issues de données réelles et modifiées algorithmiquement pour simuler la variation de paramètres tels que la distance et l'atténuation atmosphérique.

Ce chapitre s'articule comme suit. Tout d'abord nous présentons la base de données infra-rouge utilisée (section 3.3), puis nous décrivons l'algorithme étudié (section 3.4). Après avoir présenté les métriques d'évaluation de performance (section 3.5), nous mènerons l'étude des paramètres algorithmiques (section 3.6) et opérationnels (section 3.7). Nous terminerons par quelques considérations pratiques (section 3.8) et conclurons.

3.3 Base de données

Cette section présente la base de données d'images utilisée pour les études paramétriques.

3.3.1 Méthode de génération

Deux bases de données ont été générées par le CEP Arcueil, partenaire de Bertin Technologies. Les images produites sont des images hybrides [47], c'est à dire qu'elles sont issues d'images réelles modifiées algorithmiquement. Cette génération se décompose en quatre phases :

- Incrustation d'une image "haute résolution" de la cible dans une image de fond.
- Application d'un facteur de zoom décrivant la distance d'observation de la cible.
- Modification de l'histogramme de l'image pour répondre aux spécifications de contraste.
- Ajout d'un bruit électronique capteur.

L'image est définie principalement par les paramètres suivants:

- RSS, Root Sum Square: mesure de contraste local, le score est d'autant plus élevé que la cible se détache bien du fond. Cette mesure combine la différence de luminosité moyenne entre la cible et le fond ainsi que le contraste interne de la cible.
- RSC, Ratio Signal sur Clutter, le score est d'autant plus élevé que la cible se détache bien du fond. Cette mesure compare le contraste local de la cible au contraste du fond de scène.
- σ_{atmo}: caractérise la transmission atmosphérique. Plus la valeur est élevée, plus l'atmosphère transmet le signal infra-rouge et plus l'image formée sur la caméra est nette.

3.3.2 Types d'objets

La base de données contient la signature de 7 cibles différentes, qui sont représentées dans la figure 3.1. A titre d'information, les dimensions réelles de la cible 4 sont : longueur de 6.2m (9.5m) sans (avec) canon, largeur de 3.1m, hauteur de 2.5m.

Ces véhicules sont regroupés en trois classes sémantiques:

- Classe I : "Chars lourds"
- Classe II : "Véhicules blindés de transport de troupes (VBTT)"
- Classe III : "Véhicules blindés de combat d'infanterie (VBCI)"

Figure ou tableau confidentiel

Figure 3.1: Les sept véhicules vus avec une bonne résolution dans de bonnes conditions en imagerie infra-rouge. Les conditions d'observations réelles sont beaucoup plus dégradées (voir par exemple figure 3.18)

Réf	Type scénario	Nb images	Distances	RSC	RSS	K	Occultation
S	Signatures des cibles sans fond	CONFIIDENTIEL					
A	Cibles contrastée sur fond à faible clutter						
B	Cibles contrastées sur fond à fort clutter						
C	Cibles à faible contraste sur fond présentant un clutter moyen						
D	Base de test aléatoire						
E	Fond à clutter moyen						
F	Environnement à clutter moyen						

Table 3.1: Contenu de la première bases d'images hybrides du CEP: scénarios S et A à F

3.3.3 Contenu

La première base de données correspond aux signatures infra-rouge des cibles sans fond (S) (voir figure 3.1) et aux scénarios A à F. Les caractéristiques de ces scénarios sont résumées dans le tableau 3.1. Quelques images issues de ces scénarios sont présentées figure 3.18.

Une base complémentaire a été demandée au CEP à notre initiative, afin d'étudier plus en détail le comportement de l'algorithme et ses limites en fonction des principaux paramètres opérationnels (distances d'observation, atténuation atmosphérique, contraste et texture cible / fond). Le tableau 3.2 synthétise les valeurs choisies des principaux paramètres de génération pour définir trois configurations typiques (notées "Config 1" à "Config 3"). A partir de ces plages de valeurs, 45 scénarios ont été générés.

3.4 Algorithme de classification

Le but de l'algorithme proposé est de prédire la classe des objets présents dans les images qu'il doit analyser, une fois accomplie une phase d'apprentissage. Lors de l'apprentissage, l'algorithme apprend un modèle de classification à partir d'images labellisées, i.e. dont on connaît la classe. Ce modèle permet de prédire la classe d'une image non labellisée lors d'une phase de prédiction, aussi appelée phase de test.

Scénario	Sigma atmo	Distance	K	RSS	RSC	Taux occult.	Delta T cible-fond	Sigma cible	Sigma fond
A	CONFIDENTIEL								
B									
C									
D									
E									
F									
					Définition Valeur Faible				
					Définition Valeur Elevee				
			N(m,s) = normale centrée en 'm' d'écart type 's'						
H	**Principe de définition des configurations à tester :**								
Config 1	**"Difficile"**								
	Plus que B								
Config 2	**"Facile"**								
	un peu moins que A								
Config 3	**"Difficile"**								
	Un peu moins que C								
				*Valeurs **moyennes** déduites*			*Valeurs d'entrée: valeurs **moyennes** des gaussiennes*		

Table 3.2: Définition des plages des variables pour la seconde base d'images hybrides du CEP: scénario H, défini suite à l'étude paramétrique sur les scénarios A à F. On crée trois configurations, définies par des valeurs faibles ou élevées de "Delta T", "Sigma cible", "Sigma fond"

3.4.1 Principe général

Le principe général est illustré figure 3.2. On dispose de deux jeux d'images: le premier (ensemble d'apprentissage) est destiné à l'apprentissage d'un modèle de classifieur, et le second (ensemble de test) est destiné à l'évaluation de la performance du classifieur appris. Lors de l'étape 1, des données d'apprentissage sont décrites par un descripteur. Le descripteur

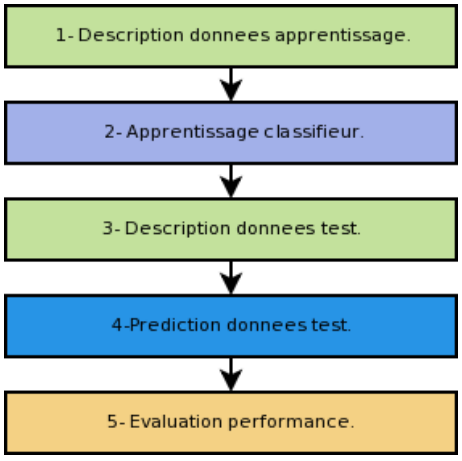


Figure 3.2: Principe général de classification

produit, à partir d'une image, un vecteur de taille fixe (la description). L'ensemble des descriptions des données d'apprentissage permet de calculer la valeur optimale des paramètres d'un classifieur (étape 2): c'est la phase d'apprentissage. Il est alors possible d'utiliser le classifieur dans un rôle de prédiction (étape 4): le descripteur utilisé à l'étape 1 est utilisé à l'étape 3 pour calculer la description d'une image de l'ensemble d'apprentissage, et le classifieur prédit sa classe. Les statistiques de prédiction des images de l'ensemble de test permettent d'évaluer la performance du classifieur (étape 5).

La partie difficile des travaux est *le calcul rapide de descripteurs efficaces*. Les descripteurs seront décrits en détails dans la partie suivante. Leur mise en oeuvre nécessite de régler de nombreux paramètres, l'étude de ces paramètres est présentée dans une autre section.

3.4.2 Le descripteur d'images

Cette partie présente en détails toutes les opérations qui permettent de transformer une image en un vecteur descripteur de taille fixe. Il existe une infinité de transformations possibles, nous présentons celle où les travaux de la thèse nous ont mené.

Les différentes étapes, qui seront détaillées plus bas, sont:

- Création d'images normalisées.
- Création d'un vocabulaire visuel. Il s'agit d'un ensemble de régions locales d'images significatives (des mots visuels) que l'on va rechercher dans les images à décrire.
- Détection des mots visuels dans les images.
- Création d'un vecteur de taille fixe: la description. Les résultats de détection de tous les mots visuels du vocabulaire visuel sont combinés pour obtenir un vecteur de taille fixe décrivant l'image.

Nous utilisons des descripteurs par niveaux de gris plutôt que SIFT (contrairement au chapitre 2) car les descripteurs SIFT et niveaux de gris donnent la même performance sur les bases de données traitées. La raison principale est la petite taille des images, la présence de bruit important et la faible résolution des images. L'avantage des descripteurs par niveaux de gris est le temps de calcul plus faible que celui des descripteurs SIFT.

Création d'images normalisées

Considérons une image issue d'un flux vidéo, comme celle présentée figure 3.3. Un détecteur de mouvement produit une région d'intérêt (ROI). Dans la base CEP, les ROIs sont données par l'algorithme de création hybride de données. Cette ROI est élargie (coefficient d'élargissement, α_{inc}[1]) dans toutes les directions afin de tenir compte de l'imprécision du

[1]Typiquement XX % et XX %

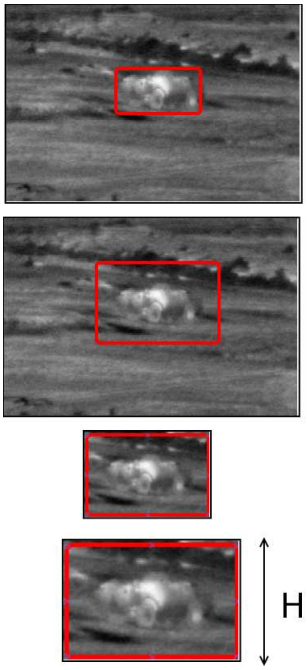


Figure 3.3: Création d'images normalisées: détection, élargissement, découpage et redimensionnement

détecteur de mouvement. Ainsi, même si la ROI détectée est un peu trop petite ou décentrée, la ROI élargie contient entièrement la partie visible de l'objet, ce qui est important pour l'algorithme de reconnaissance. Ce coefficient d'élargissement peut être déterminé en étudiant le comportement du détecteur. Si H_0 est la hauteur de la ROI d'origine[2], la hauteur de la ROI élargie est donnée par: $H_{inc} = (1+2\alpha_{inc})H_0$. Il en est de même pour la largeur. L'image est ensuite découpée autour de cette ROI: l'extérieur de la ROI est ignoré dans la suite des calculs. Enfin, elle est redimensionnée à une hauteur standard, H[3]. Cela permet de traiter indifféremment des régions de toutes tailles. Notons qu'il est plus judicieux de normaliser par la hauteur que la largeur, car la hauteur est invariante par rapport aux rotations d'objets rigides selon l'axe vertical (seul cas de rotation envisageable pour le terrestre).

Création d'un vocabulaire visuel

Un vocabulaire visuel est un ensemble de régions locales, ou mots visuels, que l'on cherche à localiser dans les images à décrire. L'utilisation du vocabulaire visuel sera détaillée plus bas, cette partie se contente de décrire le processus de création du vocabulaire visuel (voir figure 3.4). Le vocabulaire visuel est obtenu en découpant aléatoirement des régions dans

[2]De XX à XX pixels dans les scénarios fournis

[3]Typiquement XX pixels, transformation bilinéaire. $H > H_{inc} > H_0$ dans les scénarios.

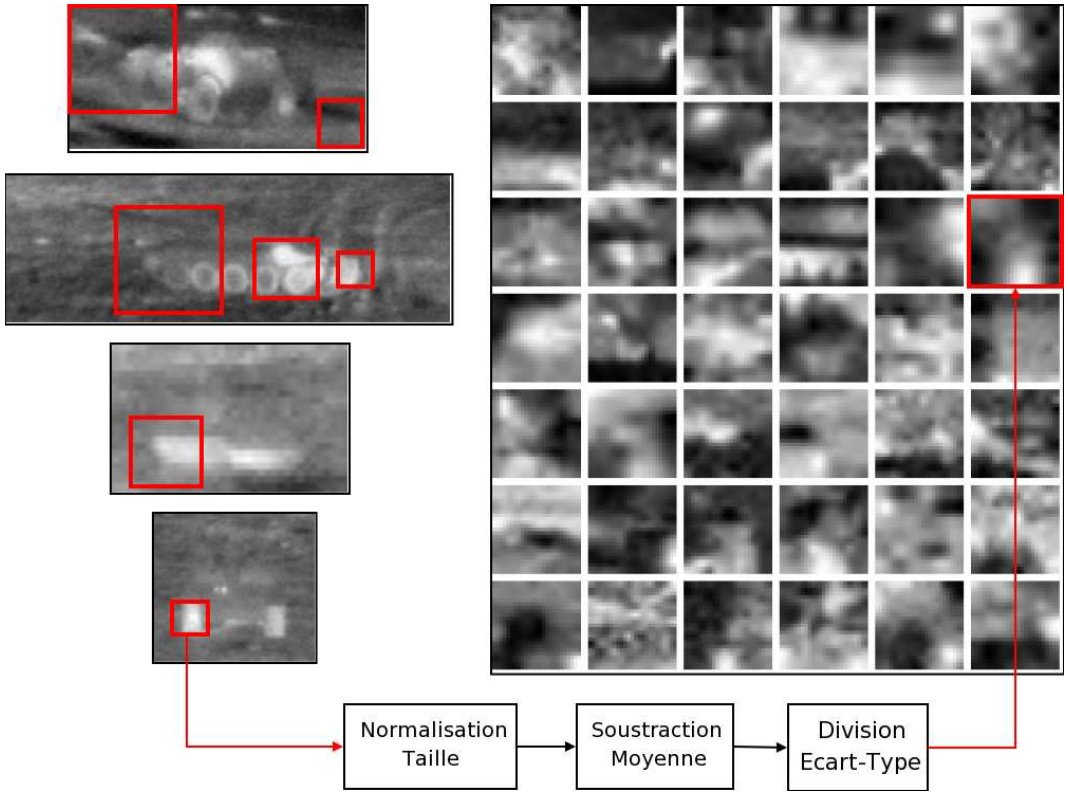


Figure 3.4: Création du vocabulaire visuel par découpage aléatoire de régions locales, puis normalisation. A droite, un exemple de (petit) vocabulaire visuel obtenu.

les images normalisées en taille par l'étape précédente, puis en normalisant les régions, comme cela est détaillé ci-après. On sélectionne D (taille du vocabulaire visuel)[4] régions carrées de tailles variant de w_{min} (taille minimum de découpage)[5] à w_{max} (taille maximale de découpage)[6] parmi les images d'apprentissage. Les images, les positions et les tailles de découpage sont choisies aléatoirement, par une loi uniforme. Les régions sont redimensionnées à une taille w_{std} (taille de région locale standard)[7].

Elles sont ensuite normalisées en luminosité et en contraste, ce qui permet d'être plus robuste aux variations locales de luminosité/contraste. Cette normalisation consiste à retrancher aux niveaux de gris des pixels leur moyenne (normalisation luminosité), puis à diviser par l'écart-type (normalisation contraste). Si l'écart type d'une région locale est

[4]Typiquement *XX*
[5]Typiquement la taille d'une roue
[6]Typiquement *XX* pixels ce n'est pas un paramètre, voir plus loin Hmin et Hmax.
[7]Typiquement *XX* pixels

inférieur à $\sigma_{uniform}$, l'écart type maximum d'une région uniforme[8] , la région locale est considérée comme uniforme (au bruit près): les niveaux de gris de tous les pixels sont mis à zéro.

Détection d'un mot visuel dans une image

Comme nous l'avons signalé, les mots visuels du vocabulaire visuel doivent être recherchés dans des images. Chaque mot visuel sera recherché à différentes positions de manière multi-échelles. Les différentes positions sont illustrées figure 3.5. Nous expliquons ci-dessous comment obtenir la liste des positions considérées (l'échantillonnage) et la mesure utilisée pour évaluer la présence d'un mot visuel aux positions échantillonnées (la distance).

L'échantillonnage Un mot visuel est recherché dans une image à différentes échelles et à différentes positions. Il est équivalent de rechercher un mot visuel de taille variable dans une image de taille fixe ou un mot visuel de taille fixe dans une image de taille variable. Nous optons pour la seconde solution, car la détection d'un petit mot visuel dans une petite image est plus rapide que la détection d'un mot visuel de grande taille.

[8]Typiquement *XX* niveaux de gris sur 256

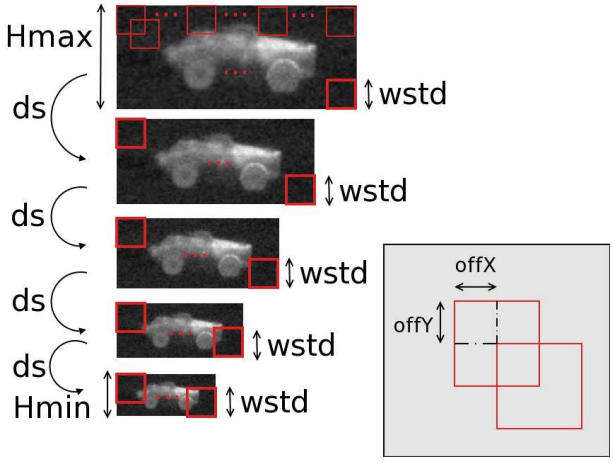


Figure 3.5: Détection multi-échelles sur grille

Nous construisons donc une pyramide d'échelles d'une image, d'une hauteur H_{max} (hauteur maximale)[9] à une hauteur H_{min} (hauteur minimale)[10] , avec un taux de sous échantillonnage ds[11]. Cela définit un nombre de niveaux ns[12] qui vérifie: $H_{min} = H_{max} * ds^{ns-1}$.

A chaque niveau d'échelle, les éléments du vocabulaire visuel sont détectés à leur taille normalisée: w_{std}. La première région considérée est la région située dans le coin supérieur gauche de l'image. Ensuite, les différentes régions considérées sont obtenues en effectuant des translations de $[k_x off_x, k_y off_y]$, pour tous les couples (k_x, k_y) d'entiers donnant des coordonnées valides. On définit $off_x = off_y = w_{std} off$, et off est l'offset de définition de la grille d'échantillonnage de détection[13].

Comme il est équivalent d'utiliser des mots visuels de taille fixe dans des images de taille variable, et des mots visuels de taille variable dans des images de taille fixe, on a la relation: $w_{std} H = w_{min} H_{max} = w_{max} Hmin$.

La distance Soit P un mot visuel du vocabulaire visuel. Soit Q l'une des régions échantillonnées dans une image. Le carré de la distance au sens mathématique du terme entre P et Q est la Normalized Sum of Square Differences, $d^2(P,Q) = NSSD(P,Q) = 2(1 - NCC(P,Q))$ où :

$$NCC(P,Q) = \frac{\sum_{i,j}(P(i,j) - \bar{P})(Q(i,j) - \bar{Q})}{\sqrt{\sum_{i,j}(P(i,j) - \bar{P})^2 \sum_{i,j}(Q(i,j) - \bar{Q})^2}} \tag{3.1}$$

où $\bar{X}$ est la moyenne des $X(i,j)$.

Si P ou Q est uniforme (écart type inférieur à $\sigma_{uniform}$), on définit: $NCC(P,Q) = 0.5$. Si P=Q, NSSD=0 et NCC=1. Si P et Q sont le plus différent possible, NSSD=4 et NCC=-1. Les moyennes $(\bar{P}, \bar{Q})$ et les écart-types (dénominateur) se calculent très rapidement à l'aide des images intégrales des intensités des pixels et de leurs carrés [90].

La distance entre un mot visuel et une image est définie comme le minimum de toutes les distances entre le mot visuel et les régions échantillonnées dans l'image. Il en découle que la NCC entre un mot visuel et une image est le *maximum* de toutes les NCC entre le mot visuel et les régions échantillonnées dans l'image.

[9] Hmax n'est pas un paramètre car il suffit de considérer que la hauteur maximale est H, la hauteur de normalisation des ROIs. Cependant, pour des besoins d'expérimentation, il est pratique de manipuler une hauteur Hmax différente de H, plutôt que de créer plusieurs bases de données d'images avec des valeurs de H différentes. Typiquement, Hmax vaut XX pixels.

[10] Typiquement XX pixels

[11] Typiquement XX

[12] Ce n'est pas un paramètre car il est entièrement déterminé par Hmin, Hmax, ds. Typiquement, ns= XX

[13] Typiquement XX

Description d'une image

L'information utilisée pour décrire une image est la NCC entre l'image et chacun des mots du vocabulaire visuel. La figure 3.6 illustre le principe avec les deux premiers mots du vocabulaire visuel. Tous les mots du vocabulaire visuel sont détectés dans l'image. Dans l'illustration, un graphe-barre représente, pour chaque mot visuel, la NCC entre le mot et l'image. On définit ensuite des seuils NCC d'activation $ncc_1, ncc_2, ..., ncc_k$[14]. Dans l'illustration, $k = 4, ncc_1 = 0.6, ncc_2 = 0.7, ncc_3 = 0.8, ncc_4 = 0.9$.

Le vecteur de description est ensuite défini par bloc, chaque bloc encodant la réponse d'un mot visuel sur l'image, et chaque élément de bloc indiquant par un 1 (ou un 0) si les seuils NCC d'activation ont été dépassés (ou non) (voir figure 3.6).

Formellement, on obtient la description suivante: $V = [v_{1,1}, v_{1,2}, ..., v_{1,k}, ..., v_{D,1}, ..., v_{D,k}]$ où $v_{i,j} = 1 \Leftrightarrow NCC(Image, Patch_i) \geq ncc_j$.

Cette description est utilisée telle quelle dans le SVM. Notons que cette information binaire est particulièrement adaptée aux SVM en raison de la haute dimensionnalité des données manipulées (i.e. vocabulaire visuel de grande taille).

3.4.3 Classifieurs SVMs

Cette partie est un rappel sur les classifieurs SVMs [88], utilisés ici pour séparer les descripteurs des différentes classes et prédire la classe d'un descripteur d'une image de test dont la classe est inconnue. Nous présentons succinctement la classification binaire et la classification multi-classes, qui en découle.

[14] Typiquement XX ... XX

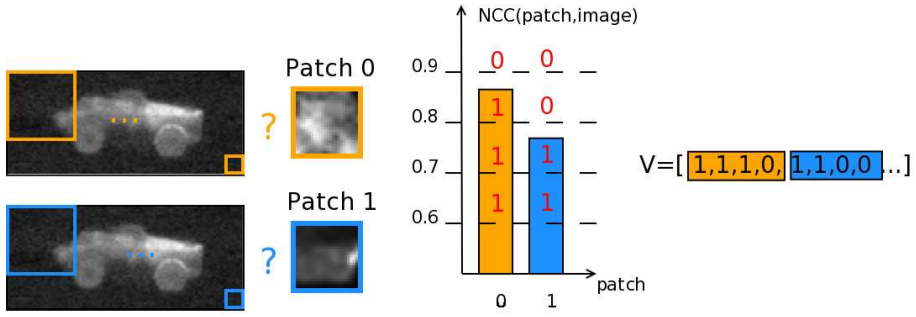


Figure 3.6: Création d'un vecteur de description d'image à partir des informations de détection de mots visuels (ou patchs)

Classification binaire

Notons $\{(x_i, y_i), i = 1...m\}$ les m données d'apprentissage. Les x_i sont les descriptions des images, ce sont des vecteurs de taille D, et les y_i sont les labels associés: +1 ou -1. Les labels indiquent à quelle classe les données appartiennent.

La classe prédite pour un exemple x inconnu est la classe +1 si

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \geq 0 \tag{3.2}$$

et la classe -1 sinon.

K est un kernel, il s'agit du produit scalaire dans le cas d'un SVM linéaire, et de la fonction $K : (x,y) \rightarrow exp(-1/2\sigma^2||x-y||^2)$ dans le cas d'un SVM RBF, avec σ une constante dont on peut estimer une valeur pratique en fonction des données d'apprentissage. b et les α_i sont des réels dont les valeurs sont optimisées grâce aux données d'apprentissage.

Dans le cas d'un classifieur linéaire, la formule de prédiction se simplifie en

$$f(x) = \omega^T x + b \tag{3.3}$$

où ω est un vecteur de taille D dont les éléments se calculent en fonction des α_i, x_i.

Classification multi-classes un-contre-un

Considérons maintenant des données dont les labels sont des entiers compris entre 1 et C, où C est le nombre de classes. On entraîne $C(C-1)/2$ classifieurs binaires, pour obtenir les fonctions de prédiction $f_{i,j}(x)$ $(i, j \in [1..C]^2, i < j)$ prédisant si un exemple appartient plutôt à la classe i ou plutôt à la classe j. Lors de la phase de prédiction, la classe qui reçoit le plus de votes de la part des $f_{i,j}$ est la classe prédite.

Temps de calcul lors de la prédiction

Le temps de calcul d'un SVM RBF est de l'odre de C^2mD, où m est le nombre d'exemples de la base d'apprentissage, C le nombre de classes, D la dimension de l'espace des données (i.e. vecteurs de descriptions). Le temps de calcul d'un SVM linéaire est de l'ordre de C^2D; contrairement au SVM RBF, le temps de calcul d'un SVM linéaire est indépendant du nombre d'exemples d'apprentissage.

3.4.4 Conclusion

Notre chaîne algorithmique consiste à décrire les ROIs qui servent à faire de l'apprentissage et de la prédiction, puis à utiliser ces descriptions pour apprendre un classifieur SVM et pour prédire la classe d'une ROI qui n'a jamais été vue lors de l'apprentissage. La description est l'étape clé de l'algorithme, ses étapes successives sont:

- la recherche de mots visuels dans des ROIs suivant une stratégie d'échantillonnage bien définie
- la construction d'un vecteur de taille fixe, appelé description, qui contient des informations sur des seuils d'activation de présence de mots visuels.

3.5 Métrique d'évaluation de performance

Avant de poursuivre, nous présentons les deux mesures de performance que nous utilisons.

La première mesure est la matrice de confusion des prédictions, voir tableau 3.3. La cellule située ligne *i* et colonne *j* contient le pourcentage d'images de la classe *i* dont on a prédit qu'ils appartiennent à la classe *j*. Dans l'exemple ci-dessus, 3% des exemples de la classe 1 sont pris pour des exemples de la classe 3. Une matrice parfaite contient des 100 sur la diagonale (et donc des 0 ailleurs): tous les éléments de chaque classe sont correctement reconnus (et donc il n'y a pas de confusion).

La seconde mesure est la moyenne des prédictions correctes, notée *meanAcc*. Il s'agit de la moyenne de la diagonale de la matrice de confusion, i.e. la moyenne des taux de reconnaissance correcte. Il faut toujours comparer ce taux au taux de réussite d'une prédiction aléatoire, qui est de 100/C, où C est le nombre de classes qu'il faut détecter. Dans l'exemple ci-dessus, meanAcc=93.3%, et le hasard est de 33.3%.

La moyenne des prédictions correctes est une information plus pauvre que la matrice de confusion, mais elle est synthétisée en une unique valeur, ce qui permet de tracer des courbes de performance en fonction de divers paramètres. Notons que dans un soucis de simplification de l'étude paramétrique, on ne prend pas en compte la variance de la diagonale.

Ces mesures tiennent compte des notions manipulées dans le domaine de la détection: vrais positifs (TP), faux positifs (FP). En effet, les cellules de la diagonale contiennent les informations: $TP = 100 - FP$, pour chaque classe.

	classe 1	classe 2	classe 3
classe 1	95	2	**3**
classe 2	3	85	12
classe 3	0	0	100

Table 3.3: Exemple de matrice de confusion. 3% des exemples de la classe 1 sont prédits comme éléments de la classe 3 (ligne 2, colonne 4, en gras).

3.6 Étude des paramètres algorithmiques

Dans cette partie, nous étudions l'influence des différents paramètres de l'algorithme présenté en détails ci-dessus. En particulier, nous nous attardons sur le compromis entre vitesse et performance. Lors de la phase de prédiction, quand il faut décrire une image puis prédire sa classe par SVM Linéaire, 95% du temps d'exécution consiste à calculer des corrélations normalisées (NCC). On peut donc considérer que toute optimisation qui diminue le nombre de détections à effectuer *diminue d'autant de temps d'exécution* de la chaîne de prédiction complète.

3.6.1 Paramètres algorithmiques

Les paramètres algorithmiques ont déjà été cités, nous les rappelons ci-dessous:

- Hauteur standard de ROI, H
- Taille du vocabulaire visuel, D
- Taille de région locale standard, w_{std}
- Écart type maximum d'une région uniforme, $\sigma_{uniform}$
- Paramètres d'échantillonnage H_{max}, H_{min}, ds
- Offset de grille d'échantillonnage , off
- Seuils NCC d'activation $ncc_1, \ldots, ncc_k$, où k est le nombre de seuils d'activation.

De plus, nous étudierons l'influence de notions qui seront introduites ultérieurement:

- La quantité d'images nécessaires à l'apprentissage.
- Le compromis du SVM.

Notons que le coefficient d'élargissement de ROI, α_{inc}, est un paramètre déduit des performances du détecteur utilisé, nous l'étudierons donc avec les paramètres opérationnels et non les paramètres algorithmiques.

3.6.2 Méthodes d'évaluation

Nous utilisons les scénarios A à D de la base d'images hybrides du CEP pour l'évaluation de la sensibilité des différents paramètres. Les scénarios A,B et C servent à l'apprentissage, et le scénario D sert à l'évaluation. Ce choix est justifié par la difficulté de la tâche, le scénario D étant considéré comme difficile en raison des occultations et de la variabilité des paramètres qui ont servi à le générer (voir tableau 3.1).

Nous considérons les 7 classes de véhicules, le taux de réussite d'une prédiction aléatoire est donc de 100/7=14%=0.14. Les taux de réussite sont présentés sous forme de nombre décimaux dans les graphes (0.14), et sous forme de pourcentages (14%) dans le texte.

3.6.3 Hauteur standard de ROI, H

Ce paramètre n'est pas étudié. L'étudier revient à mesurer la sensibilité de l'approche par rapport à la perte de résolution due au sous-échantillonnage, fait par interpolation bilinéaire.

3.6.4 Taille du vocabulaire visuel, D

Le chapitre précédent a montré que plus la taille du vocabulaire visuel augmente, plus la performance augmente. Cependant, le temps de calcul est proportionnel à la taille du vocabulaire visuel. Il faut donc trouver un compromis entre la performance et la taille du vocabulaire visuel. Nous analysons ci-dessous l'influence de la taille du vocabulaire visuel sur la performance en reconnaissance.

Pour cela, nous considérons plusieurs méthodes de classement des mots du vocabulaire visuel, qui seront largement détaillées section 4.5. Le fait de classer les mots du vocabulaire visuel du plus utile au moins utile permet de choisir les n mots les plus utiles, pour n donné. Nous considérons trois classements des mots totalement aléatoires (rand1, rand2 et rand3), qui servent de référence de base aux autres méthodes de classement. Nous considérons aussi un classement par taux d'information mutuelle (MI) et par taux d'information mutuelle conditionnelle (CondMI). Contrairement à la précédente, cette dernière considère les probabilités d'apparition de mots visuels conditionnellement aux mots visuels mieux classés, et évite ainsi l'utilisation de mots visuels trop redondants.

Les différentes figures ci dessous présentent la performance de ces méthodes de classement en gardant les n primitives jugées les plus utiles, pour n variable. Les $D-n$ autres mots visuels sont donc ignorés.

Configuration paramétrique

Les valeurs de D, H, α_{inc}, off, H_{max}, H_{min}, ds, ns, $\sigma_{uniform}$, w_{std}, $ncc_1,\ldots,ncc_k$ sont constantes (mais confidentielles) dans cette section.

Comparaison des méthodes de classement

L'observation de la figure 3.7 nous montre, premièrement, que la performance augmente avec la taille du vocabulaire visuel. La vitesse d'accroissement dépend de la méthode de classement des éléments du vocabulaire visuel. Les méthodes que nous proposons (MI, CondMI) fonctionnent bien mieux que les méthodes de classement aléatoire (rand1, rand2, rand3). On constate aussi que le classement par CondMI est un peu plus performant que le classement par MI. Ces remarques sont vérifiées pour les SVMs RBF et linéaire.

Comparaison entre SVMs Linéaire et RBF

La figure 3.8 nous permet de comparer les SVMs linéaire et RBF. Comme la théorie le prédit, le SVM RBF est plus performant dans le cas d'un vocabulaire visuel de petite taille,

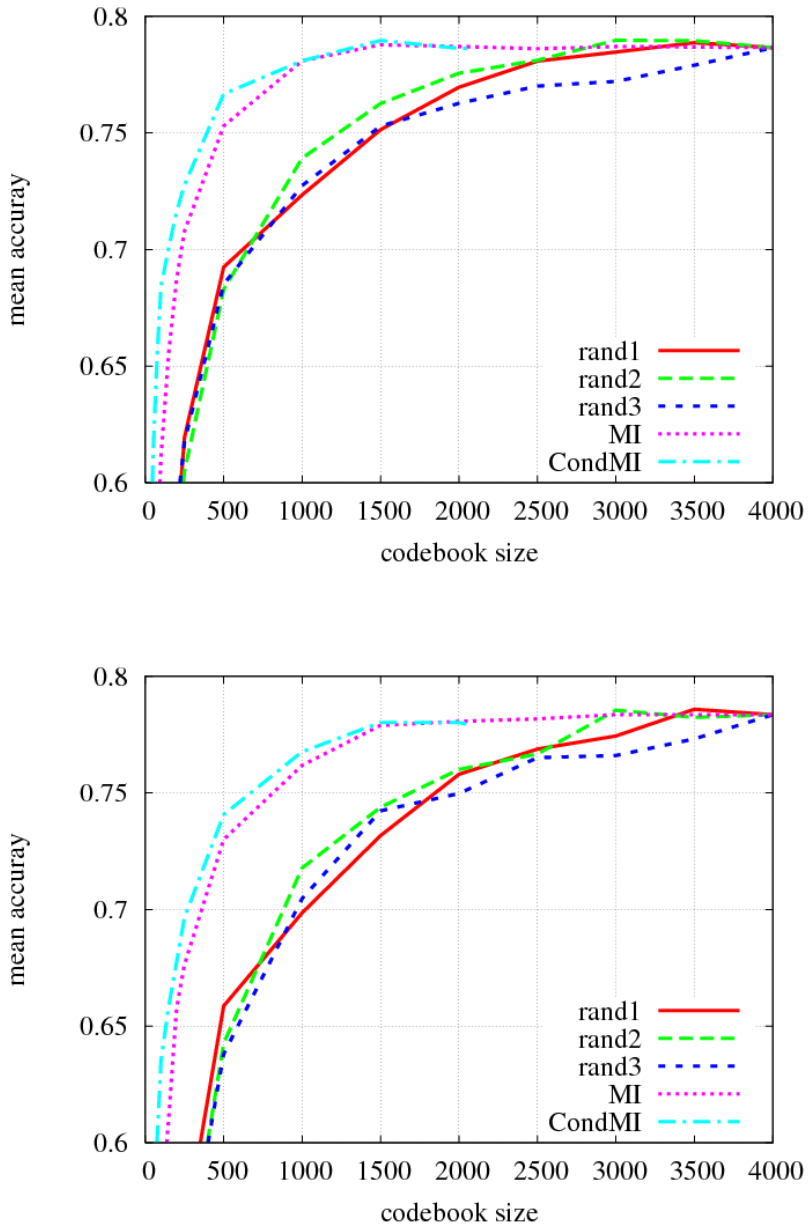


Figure 3.7: Comparaison de 5 méthodes de classement de mots visuels, avec SVM RBF (en haut) ou linéaire (en bas). En utilisant 1500 mots visuels, on atteint la performance optimale avec les méthodes de classement MI et CondMI. Les courbes représentent le taux de réussite moyen en fonction de la taille du vocabulaire.

et les différences diminuent au fur et à mesure que la taille du vocabulaire visuel augmente (comparer les courbes mauve et bleue, de même que rouge et verte). La différence varie de

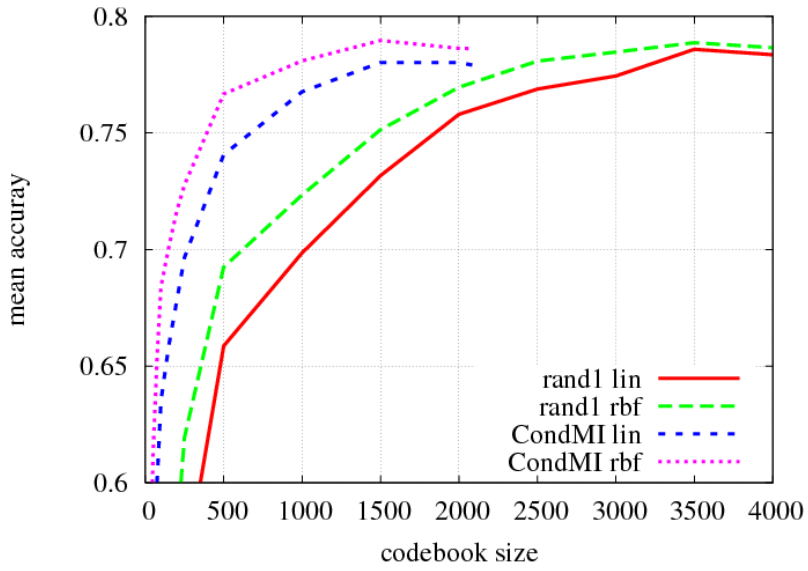


Figure 3.8: Comparaison entre les SVM linéaire et RBF. Les courbes représentent le taux de réussite moyen en fonction de la taille du vocabulaire.

0 à 3 points de pourcentage.

Les courbes mauve et bleue n'atteignent pas 4000 car la méthode de classement CondMI estime que certains mots visuels sont inutiles.

Compromis nombre de mots visuels et performance

La figure 3.9 montre le pourcentage de taux de réussite que l'on perd en fonction du pourcentage de mots visuel que l'on ignore. En réduisant la taille du vocabulaire visuel de 82.5% (on passe de 4000 à 700 mots visuels), les performances ne diminuent que de 5%. Comme le nombre d'éléments dans le vocabulaire visuel est proportionnel au temps d'exécution, cela signifie que le temps d'exécution peut être diminué de 82.5% si l'on peut supporter une diminution du taux de réussite de 5%.

Prédictivité du comportement

La courbe de la figure 3.10 montre qu'il est possible de prévoir le comportement de la réduction de la taille du vocabulaire visuel à partir des images d'apprentissage. En effet, ce n'est pas tout d'observer le comportement du compromis taille/performance sur l'ensemble de test, il est préférable de prédire ce comportement à partir des exemples d'apprentissage uniquement.

On considère pour cela un classifieur bayésien naïf, dont la propriété est de ne pas faire de sur-apprentissage. On utilise ce classifieur en validation croisée pour les méthodes de

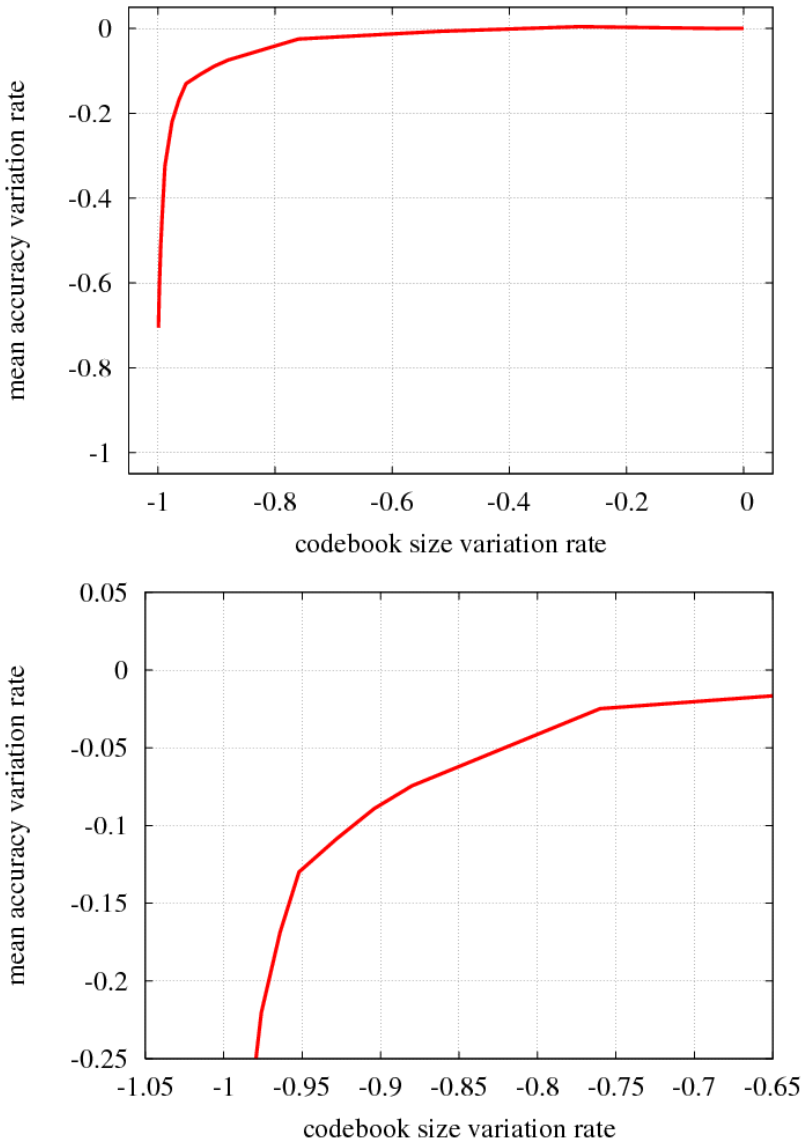


Figure 3.9: SVM RBF et classement par CondMI: variation du taux de performance en fonction de la variation de la taille du vocabulaire visuel. Haut: graphe complet, bas: zoom

classement de mots visuels rand1 et CondMI, avec les données d'apprentissage. On utilise aussi un classifieur SVM RBF sur les données de test. On constate que les deux classifieurs suivent approximativement la même évolution (courbes rouge et verte, de même que bleue et mauve).

Notons que l'évaluation des performances en validation croisée sur A,B,C donne des résultats bien meilleurs que l'évaluation de la performance sur D. Cela s'explique en consid-

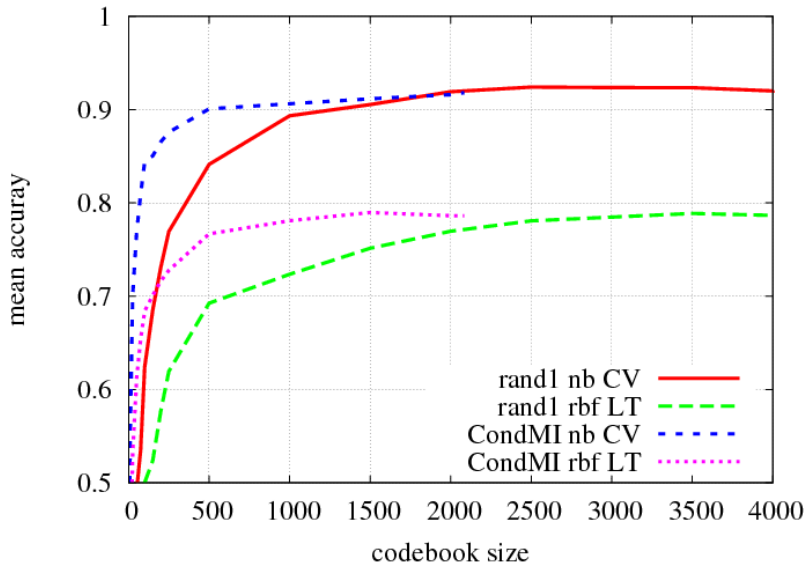


Figure 3.10: Comparaison entre le taux de réussite en validation croisée sur ABC par classifieur bayésien naïf et du taux de réussite avec apprentissage sur ABC et test sur D, avec SVM RBF. Les courbes représentent le taux de réussite moyen en fonction de la taille du vocabulaire.

érant que l'évaluation en validation croisée permet de mesurer la performance en conditions de test similaires à celles de l'apprentissage. C'est pour cette raison que la performance diminue sur le scénario D: les données sont générées par des paramètres très différents des scénarios A,B,C (occultations par exemple).

Conclusion

Nous avons vu que le classement de mots visuels par taux d'information mutuelle conditionnelle (CondMI) associé à une classification par SVM RBF donne les meilleurs résultats, c'est à dire que le compromis entre le nombre de mots visuels (donc la vitesse) et la performance est toujours meilleur avec ce choix de classifieur et de méthode de sélection de primitives.

Il est important de noter que la tendance du comportement sur les données de test est prédictible à partir des données d'apprentissage. On peut donc déterminer la taille de vocabulaire visuel optimale (1500 ici) à partir des données d'apprentissage, quelques soient les données à traiter.

3.6.5 Taille de région locale standard, w_{std}

Nous n'avons pas réalisé d'étude sur la taille w_{std}. Étant données les expériences menées sur le visible, nous pensons que le comportement est similaire à l'illustration figure 3.11, et que la configuration actuelle ($w_{std} = XX$ pour $H_{max} = XX$) se situe à droite du seuil critique.

3.6.6 Écart type maximum d'une région uniforme $\sigma_{uniform}$

Nous n'avons pas réalisé d'étude sur $\sigma_{uniform}$. Étant données les expériences menées sur le visible, nous pensons que le comportement est similaire à l'illustration figure 3.12, et que la configuration actuelle ($\sigma_{uniform} = XX$) se situe entre les deux seuils critiques.

3.6.7 Paramètres d'échantillonnage H_{max}, H_{min}, ds

Ces paramètres permettent de définir le rapport entre le côté de la région locale standard w_{std} et la hauteur h ($H_{min} \leq h \leq H_{max}$) des images dans lesquelles les mots visuels sont détectés.

L'intérêt d'analyser ces paramètres est multiple. Tout d'abord, cela nous permet de déterminer les conditions de fonctionnement optimal de l'algorithme. De plus cela nous permet d'envisager des optimisations au niveau du temps de calcul, car moins il y a de niveaux et plus les hauteurs d'images manipulées sont faibles, plus le processus de détection est rapides. Rappelons que le temps de calcul est proportionnel au nombre de positions échantillonnées.

Configuration paramétrique

Les valeurs de D, H, α_{inc}, off, $\sigma_{uniform}$, w_{std}, $ncc_1, \ldots, ncc_k$ sont constantes (mais confidentielles) dans cette section.

Dans cette section, nous mesurons l'influence de la taille relative des régions locales par rapport aux hauteurs des ROIs de la pyramide d'échelles, $w_{rel} = w_{std}/h$.

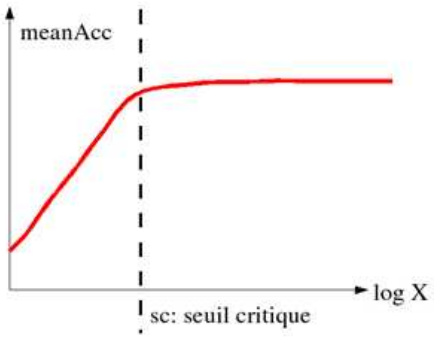


Figure 3.11: Taux de réussite en fonction du log de w_{std} (estimation intuitive)

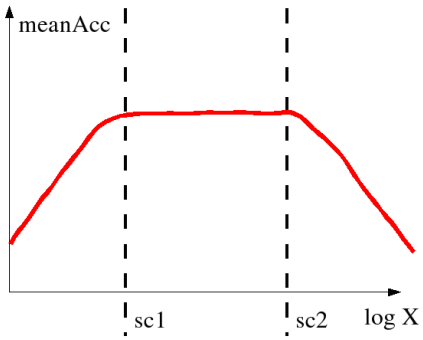


Figure 3.12: Taux de réussite en fonction du log de $\sigma_{uniform}$ (estimation intuitive)

Performance avec une échelle unique

L'illustration figure 3.13 est obtenue en utilisant un unique niveau d'échelle lors de la phase de détection. La configuration paramétrique correspondante est $H_{max} = H_{min} = w_{std}/w_{rel}$ pixels, $ns = 1$.

On constate que si on n'utilise qu'une échelle, la performance est stable et élevée ($meanAcc > 0.775$) avec des tailles de régions locales relatives variant de 14% à 35% de la hauteur de l'image. Cela correspond à l'idée que l'on se fait d'une représentation par parties: les

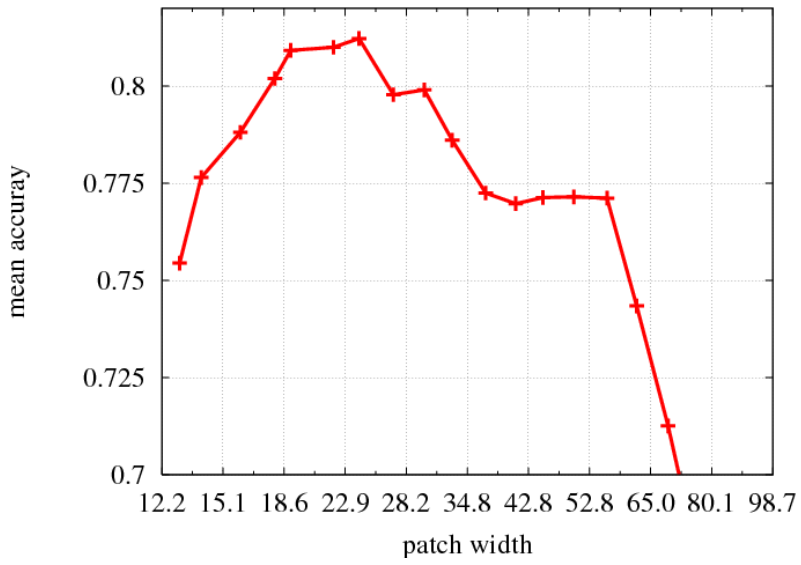


Figure 3.13: Taux de réussite en fonction de la taille relative (en pourcentage) région locale/hauteur d'image, si on n'utilise qu'une échelle pour la détection.

parties ne doivent pas être trop petites (on obtient du bruit) ni trop grandes (on obtient une représentation locale). La performance maximale est atteinte avec une taille relative de 24%, la performance est alors de 81.2% de réussite.

Combinaison d'échelles

Considérons maintenant la combinaison d'informations issues de plusieurs échelles. La combinaison des vecteurs de tailles $w_{rel} = 24\%$ et $w_{rel} = 45\%$ par un OU logique[15] donne une performance de 79.8%, ce qui est moins bon qu'une seule taille. Une combinaison des vecteurs par concaténation[16] donne une performance de 81.3%, ce qui est un peu meilleur qu'une seule taille, mais de manière non significative.

Ensemble d'échelles

Dans le cas de la base hybride, les ROIs sont toujours découpées selon le même principe déterministe: localisation de la vérité terrain, élargissement, découpage, redimensionnement. Il est donc possible de n'utiliser que la taille relative la mieux adaptée, car on connaît précisément la taille de l'objet. Cependant, lors d'une utilisation réelle, le coefficient d'élargissement

[15] Si l'une des tailles est détectée, le mot visuel correspondant est mis à 1

[16] Il y a deux dimensions par mot visuel: une pour chaque taille

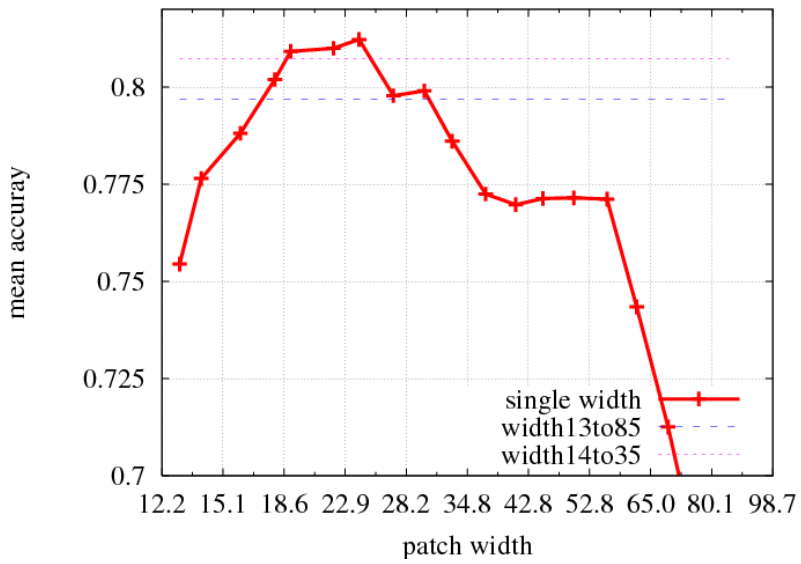


Figure 3.14: Taux de réussite en fonction de la taille relative (en pourcentage) région locale/hauteur d'image : comparaison entre les tailles uniques et les intervalles de tailles

varie d'une ROI à l'autre, il n'y a donc pas de meilleure échelle relative. Cette section analyse s'il est préférable de ne considérer qu'une échelle, ou d'utiliser un ensemble d'échelles.

Nous considérons deux ensembles de tailles relatives. Le premier, E_1, est constitué des valeurs 13, 14, 16, 18, 19, 22, 24, 27, 30, 33, 37, 41, 45, 50, 56, 62, 69, 77 et 85. Le second, E_2, est constitué des valeurs 14, 16, 18, 19, 22, 24, 27, 30 et 33. Ce sous-ensemble de E_1 correspond à l'intervalle où les performances avec échelle unique sont supérieures à 77.5%. L'écart entre deux valeurs correspond à un taux de sous-échantillonnage de 0.9.

Les performances des différentes approches sont présentées sur la figure 3.14. On constate que la meilleure performance (81.2%) est obtenue pour une taille relative de 24%. Cependant, si la taille relative est plus grande ou plus petite, la performance devient moins bonne que celle obtenue avec l'ensemble réduit E_2 (80.7%). Avec des tailles de 18% ou de 30%, la performance décroît et est du même ordre que celle obtenue avec l'ensemble large E_1 (79.7%). Il est donc plus prudent d'utiliser un intervalle réduit autour de la valeur optimale plutôt que d'utiliser une unique valeur proche de la valeur optimale. Sauf si la priorité est l'accélération des calculs, au dépens d'une perte de performance.

Taux de sous-échantillonnage *ds*

Dans le cas d'utilisation d'un ensemble de tailles relatives plutôt que d'une valeur unique, il convient de se demander quel est le taux de sous-échantillonnage à adopter.

Par exemple, avec un taux de sous-échantillonnage de 0.9, l'intervalle E_2 est constitué des valeurs 13, 14, 16, 18, 19, 22, 24, 27, 30, 33, 37, 41, 45, 50, 56, 62, 69, 77 et 85. Avec un taux de sous-échantillonnage de 0.75, il ne contiendrait plus que les valeurs 13, 17, 23, 31, 41, 55, 73. Il y aurait donc 2.7 fois moins de détections à calculer.

Nous n'avons pas étudié le compromis entre le taux de sous-échantillonnage et le taux de réussite.

Conclusion

La performance optimale s'obtient en utilisant une seule échelle. Cependant, cette échelle est un paramètre sensible. En utilisant une gamme d'échelles plutôt qu'une seule échelle, la performance diminue un peu mais les paramètres d'échelle deviennent peu sensibles.

3.6.8 Offset de grille d'échantillonnage, *off*

Plus l'offset de définition de la grille d'échantillonnage des détections est élevé (voir figure 3.5), moins il y a de positions échantillonnées et plus l'algorithme est rapide. Dans cette partie, nous évaluons le compromis offset/performance.

Configuration paramétrique

Les valeurs de D, H, α_{inc}, H_{max}, H_{min}, ds, ns, $\sigma_{uniform}$, w_{std}, $ncc_1, \ldots, ncc_k$ sont constantes (mais confidentielles) dans cette section. Nous étudions l'influence de la variable off.

Influence de la variable off

Le nombre de mots visuels échantillonnés est proportionnel au carré de l'offset. La figure 3.15 permet d'observer la performance en fonction de l'offset, ainsi que le gain en temps de calcul. Avec un offset minimum ($1/w_{std}$), la performance est de 82.3%. Avec un offset de 1, la performance est de 76.8%, pour des calculs 144 fois plus rapides.

3.6.9 Seuils NCC d'activation $ncc_1, ncc_2, \ldots, ncc_k$

Le but de cette partie est d'étudier l'influence de la valeur et de la quantité des seuils NCC d'activation.

Le nombre de seuils k est indépendant du nombre de corrélations calculées, et de ce fait influence de manière négligeable le temps d'exécution de l'algorithme de prédiction. Le temps de calcul de la prédiction de la classe d'une image une fois sa description calculée est proportionnelle à k, mais le calcul est si rapide par rapport au calcul de la description qu'il a peu d'influence dans la chaîne complète.

Configuration paramétrique

Les valeurs de D, H, α_{inc}, off, H_{max}, H_{min}, ds, ns, $\sigma_{uniform}$, w_{std} sont constantes (mais confidentielles) dans cette section. Les variables étudiées sont le nombre de valeurs d'activations, et leurs valeurs.

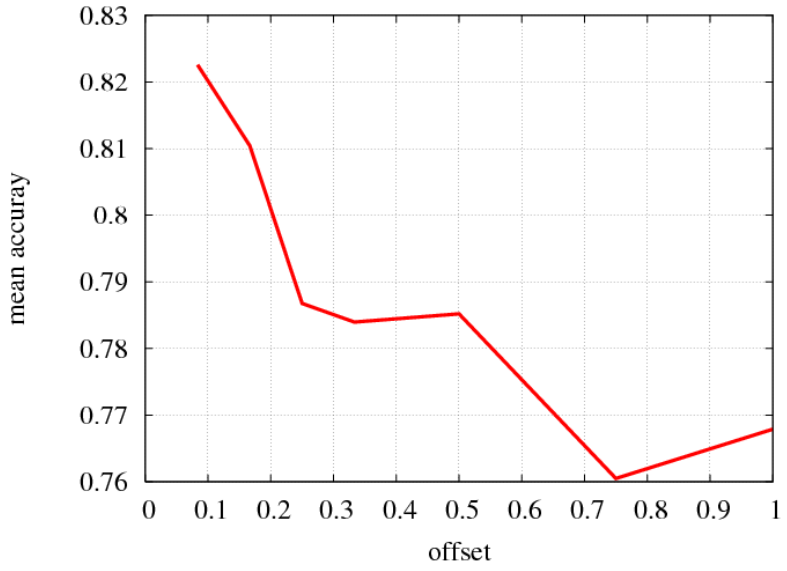


Figure 3.15: Performance en fonction de l'offset.

Influence du nombre de seuils et de leurs valeurs

La figure 3.16 montre l'intérêt d'utiliser les 4 seuils 0.65, 0.7, 0.75, 0.8 plutôt que chacun d'entre eux individuellement. En utilisant les 4 seuils, la performance est de 80.3%. En utilisant un seul seuil, la performance diminue de 1.9 à 3.8 points de pourcentage.

Conclusion

L'utilisation de plusieurs seuils d'activation permet, sans surcharge de calculs, d'améliorer la performance.

3.6.10 Quantité d'images nécessaires à l'apprentissage

Le principe d'un classifieur n'est pas de reconnaître précisément ce qui a été vu lors d'une phase d'apprentissage, mais de généraliser afin de reconnaître des formes similaires mais différentes. Plus le classifieur dispose d'images représentatives lors de l'apprentissage, meilleure est sa performance lors de la prédiction. Cette section analyse la performance en fonction du nombre d'exemples vus, par catégorie d'objet, lors de l'apprentissage.

Notons cependant que si les formes vues lors de l'apprentissage et de la prédiction sont trop différentes, la prédiction est alors aléatoire. Par exemple, si un véhicule est appris de face mais que le véhicule est vu de profil lors de la prédiction, celle-ci devient aléatoire, car l'apprentissage n'apporte pas d'information. De même, si un véhicule est vu par temps

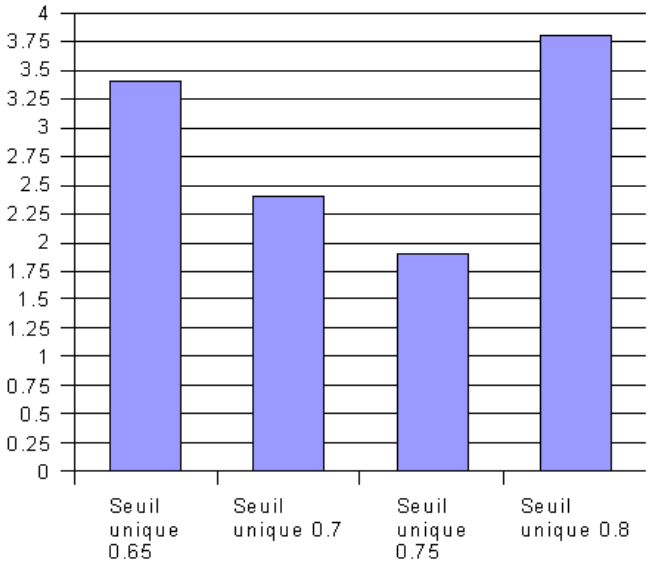


Figure 3.16: Nombre de points de pourcentage de taux de réussite perdus par rapport à l'utilisation des 4 seuils simultanément

chaud et sec de près lors de l'apprentissage mais qu'il est vu de loin par temps froid et humide lors de la prédiction, celle-ci devient aléatoire.

Configuration paramétrique

Les valeurs de D, H, α_{inc}, off, H_{max}, H_{min}, ds, ns, $\sigma_{uniform}$, w_{std}, $ncc_1, \ldots, ncc_k$ sont constantes (mais confidentielles) dans cette section.

Cette section étudie la performance en fonction du nombre d'exemples n utilisés, par classe, lors de l'apprentissage. Les n exemples sont tirés aléatoirement parmi toutes les images de la classe en question dans les scénarios A,B et C. Il est important de noter qu'on ne tire pas n exemples par configuration paramétrique (classe 1 + distance courte + RSC faible + RSS élevé + occultation faible + ...) , mais réellement n images tous paramètres autres que la classe confondus.

Influence du nombre d'images d'apprentissage

La figure 3.17 montre la performance sur la base de test D (courbe rouge) en fonction du nombre d'exemples utilisés, par classe, lors de l'apprentissage. La performance augmente

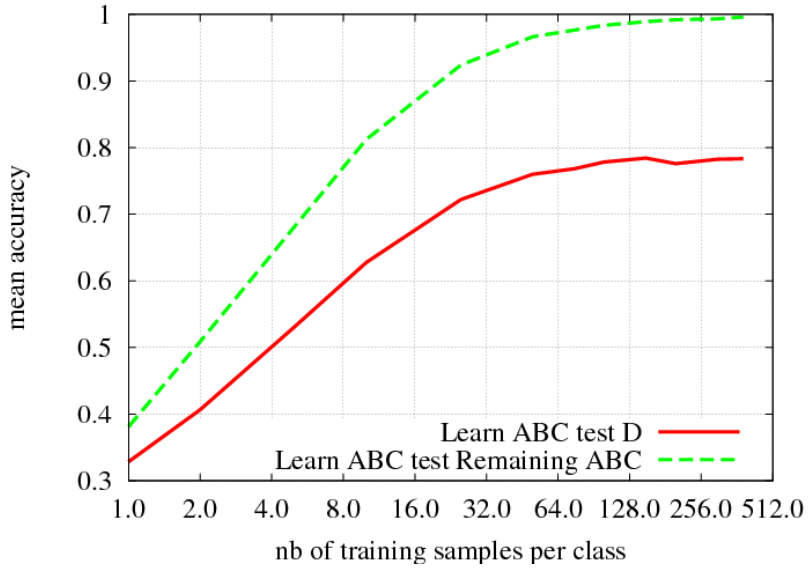


Figure 3.17: Taux de réussite en fonction du nombre d'images utilisées, par classe, lors de l'apprentissage. La courbe verte représente une situation de "déjà-vu": toutes les données sont générées par la même procédure, et la performance est mesurée par validation croisée. La courbe rouge représente une situation de "jamais vu": les données sont apprises sur les bases A,B,C et la performance est testée sur la base D, qui contient plus de bruit et des occultations.

linéairement en fonction du log du nombre d'images jusqu'à 32 images par classe. Ensuite l'augmentation ralentit, et entre dans un comportement asymptotique à partir de 128 images par classe.

La courbe verte montre la performance sur les images de la base A,B,C non utilisées lors de l'apprentissage. Les données non-utilisées sont générées par le même processus, mais sont tout de même différentes. On constate premièrement que les performances sont meilleures que celles observées sur la base D, ce qui est normal car la base D est générée par un processus différent (présence d'occultations par exemple). On constate aussi que les deux courbes suivent la même tendance, on peut donc prédire le comportement en situation de test à partir des données d'apprentissage.

Conclusion

Prenons le cas de seize images utilisées, par classe, pour l'apprentissage. Si on considère qu'il y a huit orientations caractéristiques , et que les autres paramètres opérationnels se résument à deux configurations (une facile et une difficile), cela signifie qu'avec en moyenne une seule image par combinaison orientation/niveau de difficulté, l'algorithme a une performance de 68% sur la base D, et 87% sur la base A,B,C, alors que le hasard est de 14%. On peut donc conclure que la rapidité d'apprentissage et la capacité à généraliser sont exceptionnelles.

3.6.11 Compromis d'apprentissage du SVM

Il s'agit de l'unique paramètre à régler manuellement pour utiliser un SVM linéaire. Pour un SVM RBF, il y en a un autre, mais on peut déduire sa valeur à partir des données.

Il est très difficile de prédire la valeur optimale de ce paramètre à partir des données d'apprentissage, d'autant plus si les données de test sont générés par un processus différent. Nous avons mesuré la différence entre la performance maximale et la performance obtenue pour une valeur standard (1), en apprenant sur les bases A,B,C et en testant sur la base D, ainsi qu'en apprenant sur la base A et en testant sur la base B. La différence varie de 0 à 0.5%. On utilise donc la valeur de C standard.

3.6.12 Conclusion de l'étude algorithmique

Nous connaissons les valeurs (quasi) optimales des paramètres suivants:

- taille du vocabulaire visuel (D): 1500. Cette valeur peut être déterminée automatiquement à partir des données d'apprentissage.
- échantillonnage (H_{min}, H_{max}): tailles relatives région locale/hauteur d'image entre XX et XX . Cela correspond à l'idée générale d'une détection d'objets par parties.
- taux de sous échantillonnage (ds): le plus faible possible (XX par exemple).

- offset de grille d'échantillonnage (off_x, off_y): de manière à avoir $off_x = off_y = 1$.
- seuils d'activation ($ncc_1, ..., ncc_k$): nombreux (par exemple, XX ... XX).
- nombre d'exemples par classe pour l'apprentissage: 128 (avec les paramètres opérationnels des scénarios A,B,C,D). Cette valeur peut être déterminée automatiquement à partir des données d'apprentissage.
- type de classifieur: SVM RBF (plus performant que SVM linéaire d'une manière générale, ce qui a été vérifié dans nos expérimentations).
- compromis du SVM: 1 (valeur standard).

Nous n'avons pas encore effectué une étude sur les paramètres suivants, mais nous pensons que les valeurs actuelles sont proches des valeurs optimales:

- taille de région locale standard (XX pixels de côté)
- écart-type maximum d'une région uniforme (XX)

Dans le but d'obtenir la performance optimale sans contrainte de temps, les valeurs recommandées des paramètres sont celles données ci-dessus. Par contre, dans le but d'avoir un bon compromis performance/temps de calcul, il convient de choisir des valeurs moins optimales. Le compromis dépend des performances attendues, nous donnons ci-dessous des exemples de gain en temps de calcul:

- vocabulaire visuel de taille 1000 au lieu de 4000: perte de 2 points de pourcentage, calculs 4 fois plus rapides.
- échantillonnage avec une seule taille (XX), au lieu d'un ensemble de tailles allant de XX à XX avec un taux de sous-échantillonnage de 0.9: calculs XX fois plus rapides, instabilité sur les extractions des ROIs peu précises.
- offset de grille d'échantillonnage passant de XX à XX : perte de XX points de pourcentage, calculs XX fois plus rapides.
- classifieur linéaire au lieu de RBF: perte de 2 points de pourcentage, calculs un peu plus rapides (non quantifié, gain probablement inférieur à 1.1).

3.7 Étude des paramètres opérationnels

Dans cette section, notre but est d'étudier la stabilité de l'algorithme proposé face à des variations de paramètres opérationnels (distance cible-caméra, RSC, etc.).

Nous menons deux études. La première consiste à évaluer la sensibilité des paramètres en apprenant un classifieur dans des conditions faciles, et en évaluant la performance sur

des données plus difficiles. Nous nommons cette étude *stabilité jamais-vu*. La seconde consiste à faire de l'apprentissage et de l'évaluation sur des données de même difficulté, pour différents niveaux de difficulté. Nous nommons cette étude *stabilité déjà-vu*.

Afin de faciliter notre étude paramétrique, nous discrétisons les différentes variables. Par exemple, nous définissons une distance comme faible si elle est inférieure à XX mètres, élevée si elle est supérieure à XX mètres et moyenne sinon. Ainsi, au lieu de manipuler une variable continue, on manipule une variable discrète ne pouvant prendre que 3 valeurs. Les différentes variables opérationnelles (variables terrain) considérées par cette étude sont:

- la distance cible / caméra (en mètres)
- le taux d'occultation de la cible (en pourcentage de pixels occultés)
- le RSC de la cible (ratio signal sur clutter)
- le RSS de la cible (rootsum square)
- précision de détection des ROIs, modélisées par le coefficient d'élargissement des ROIs dans chaque direction α_{inc}

Nous étudions aussi l'effet des regroupements de classe: 7 classes, ou 3 groupes de classes, ou 2 groupes de classes.

3.7.1 Discrétisation des variables

Nous définissons des intervalles de valeurs pour les différents paramètres opérationnels. Les intervalles sont définis en considérant les valeurs minimales et maximales des scénarios A,B, C et D, ainsi que la signification opérationnelle des intervalles. Les intervalles sont définis dans le tableau 3.4.

Paramètre	*Faible*	*Moyen*	*Élevé*	*Difficulté*
Distance (m), notée *dist.*	CONFIDENTIEL			croissant
Occultation (%), notée *occ.*				croissant
RSC, notée *RSC.*				décroissant
RSS, notée *RSS.*				décroissant
α_{inc}, notée *inc.*				croissant

Table 3.4: Discrétisation des paramètres opérationnels. La dernière colonne indique si la difficulté croît ou décroît quand les valeurs de paramètres varient de faible à élevé

3.7.2 Configuration des paramètres algorithmiques

Les valeurs de D, H, α_{inc}, off, H_{max}, H_{min}, ds, ns, $\sigma_{uniform}$, w_{std}, $ncc_1, \ldots, ncc_k$ sont constantes (mais confidentielles) dans cette section.

3.7.3 Stabilité en situation de jamais-vu

Parmi les scénarios A,B,C et D, nous sélectionnons tous les exemples faciles pour l'apprentissage. Ceux-ci sont définis par:

- distance faible
- occultations faible
- RSC élevé
- RSS élevé
- erreur de détection des ROIs faible

Les configurations des images d'évaluation sont identiques, hormis pour les propriétés spécifiées dans le tableau 3.5. A la vue de ce tableau, on peut conclure qu'en situation de jamais-vu, l'algorithme est très stable sauf:

- RSC faible (performance de *XX*)
- RSC faible avec une grande imprécision d'extraction de ROI (performance de *XX*)
- RSC moyen avec un taux d'occultation élevé (performance de *XX*)

Modification par rapport à la configuration facile	Taux de réussite
RSC moyen	*XX*
inc élevé	*XX*
dist moyen	*XX*
dist élevé	*XX*
RSC moyen + RSS faible	*XX*
RSC moyen + inc élevé	*XX*
RSC faible	*XX*
RSC faible + inc élevé	*XX*
RSC moyen + occ élevé	*XX*

Table 3.5: Performance avec apprentissage en conditions faciles, et évaluation en conditions faciles sauf pour les paramètres mentionnés dans la colonne de gauche

On peut expliquer les deux premiers points par la présence d'un fond très texturé, dont certaines parties peuvent ressembler à des parties de véhicules. Ces parties sont d'autant plus nombreuses que la ROI est extraite avec une faible précision. Comme des parties de fond ressemblant à des parties d'objet sont intégrées lors du calcul de la description, celle-ci est bruitée. La solution pour palier ce problème est d'utiliser le masque de détection précis de l'objet afin de n'intégrer aucune information du fond.

On peut expliquer le troisième point par l'impossibilité de détecter des parties d'objets, ce qui rend la description incomplète. Ce point n'est pas réellement un problème car il est facile de simuler des occultations lors de l'apprentissage, ce qui permet à l'algorithme d'apprendre à reconnaître des formes occultées. De plus, l'intégration d'informations temporelles peut aider à diminuer les problèmes dus aux occultations.

Différents cas de figure sont illustrés figure 3.18.

3.7.4 Stabilité en situation de déja-vu

Afin de mesurer la sensibilité de l'algorithme face à une situation où les données d'apprentissage sont similaires aux données de test, nous effectuons une évaluation par validation croisée dans des conditions homogènes.

Par exemple, on considère les images faciles hormis le fait que la distance soit élevée (conditions faciles sauf une condition difficile). On divise ce groupe en 3, on utilise les 2 premiers pour faire de l'apprentissage et le troisième pour l'évaluation, puis le premier et le troisième pour l'apprentissage, puis les deux derniers pour l'apprentissage. La performance reportée est la moyenne des trois évaluations. Chaque image est donc utilisée une et une seule fois pour l'évaluation de la performance.

Les configurations de chaque ensemble de test est la configuration facile, hormis pour les propriétés spécifiées dans le tableau 3.6. A priori, il peut sembler étrange que les configurations difficiles donnent des performances meilleures que les performances faciles. Par exemple, si on considère toutes les distances, la performance (*XX*) est meilleure que si on ne considère que les distances moyenne et élevée (*XX*)). La raison de cette différence est le nombre d'images utilisées lors de l'apprentissage. L'augmentation de cette quantité suffit, en améliorant la capacité à généraliser, à compenser l'augmentation de la difficulté. Cette considération prise en compte, on constate qu'il ne subsiste, dans une situation de déjà-vu, qu'une seule difficulté: les occultations. Cependant, cette difficulté est moindre qu'en cas de jamais-vu.

Figure ou tableau confidentiel

Figure 3.18: Différentes configurations du véhicule 12 vu à une orientation de 45 degrés. Ligne 1: configuration facile - Ligne 2: RSC faible et bonne précision d'extraction - Ligne 3: RSC faible et mauvaise précision d'extraction - Ligne 4: RSC faible et occultation élevée

Modification par rapport à la configuration facile	Taux de réussite
dist faible+moyen+élevé	*XX*
RSC moyen+élevé + RSS moyen+élevé	*XX*
RSC faible+moyen+élevé	*XX*
dist faible+moyen	*XX*
inc faible+élevé	*XX*
RSC moyen+élevé	*XX*
RSC faible+moyen+élevé + inc faible+élevé	*XX*
RSC moyen+élevé + inc faible+élevé	*XX*
Configuration facile	*XX*
RSC moyen+élevé + occ faible+élevé	*XX*

Table 3.6: Performance en validation croisée en conditions faciles, sauf pour les paramètres mentionnés dans la colonne de gauche

3.7.5 Nombre de classes

Analysons maintenant la performance en fonction du nombre de classes.

On considère:

- les 7 classes de véhicules présents dans la base de donnée CEP
- les 3 groupes de classes définis par le CEP:
  - I : chars lourds
  - II : véhicules blindés de transport de troupes, VBTT
  - III: véhicules blindés de combat d'infanterie = VBCI
- les 2 groupes de classes définis par Bertin:
  - I: *XX* ... *XX*
  - II: *XX* ... *XX*

Les performances d'apprentissage en fonction des groupes cibles sont indiquées dans le tableau 3.7. On constate que, naturellement, la performance augmente quand le nombre de groupes diminue. Cela s'explique si on considère que la confusion entre objets disparaît si les objets sont fusionnés en un groupe, d'autant plus si le groupe contient des objets qui portent à confusion, c'est à dire des objets similaires.

Aujourd'hui, nous ne disposons d'aucun élément pour prédire le comportement de l'algorithme avec un nombre de classes élevé (20 par exemple). Notons que l'algorithme actuel doit impérativement prédire l'une des C classes définies, il n'existe pas de classe rejet.

Nombre de groupes	Taux de réussite	Hasard
2	*XX*	50
3	*XX*	33.3
7	*XX*	14.3
20	*XX*	5

Table 3.7: Performance en fonction du nombre de classes. Il serait intéressant de disposer de 20 classes pour observer l'évolution du taux de réussite.

3.7.6 Influence de σ_{atmo}

Les conditions d'apprentissage sont les suivantes:

- configuration RSC/RSS difficile (configuration 1)
- σ_{atmo} bon (valeur de *XX*)
- distance courte (*XX*)

Les conditions d'évaluation sont les suivantes:

- configuration RSC/RSS moyenne (configuration 3)
- toutes valeurs de σ_{atmo}
- toutes distances

Les performances sont présentées figure 3.19. On en conclut que la performance diminue avec le sigma atmosphérique, d'autant plus que la distance augmente.

3.7.7 Influence des configurations RSS/RSC

Les conditions d'apprentissage sont les suivantes:

- configuration RSC/RSS difficile (configuration 1)
- σ_{atmo} bon (valeur de *XX*)
- distance courte (*XX*)

Les conditions d'évaluation sont les suivantes:

Figure ou tableau confidentiel

Figure 3.19: Performance en fonction de la distance pour diverses valeurs de σ_{atmo}

- toutes configurations RSC/RSC
- σ_{atmo} bon (valeur de XX)
- toutes distances

Les performances sont présentées figure 3.20. On en conclut qu'aux distances où la performance est correcte, il y a peu de différences entre les différentes configurations RSS/RSC.

3.7.8 Influence des conditions d'apprentissage

Les conditions d'apprentissage sont les suivantes:

- configuration RSC/RSS difficile (configuration 1)
- σ_{atmo} bon (valeur de XX)
- distance courte (XX)

Les conditions d'évaluation sont les suivantes:

- configuration RSC/RSS facile (configuration 2)
- σ_{atmo} bon (valeur de XX)
- toutes distances

Les performances sont présentées figure 3.21. On en conclut que plus on dispose d'informations lors de l'apprentissage, meilleure est la performance (à XX , quand on ajoute de l'information, la performance passe de XX % à XX % puis XX %, quand le hasard obtient 14%). Notons qu'il est plus intéressant de connaître plusieurs conditions atmosphériques que plusieurs distances lors de l'apprentissage.

3.7.9 Influence de la distance

Les conditions d'apprentissage sont les suivantes:

- configuration RSC/RSS difficile (configuration 1)
- toutes valeurs de σ_{atmo}

Figure ou tableau confidentiel

Figure 3.20: Performance en fonction de la distance pour diverses configurations RSS/RSC

Figure ou tableau confidentiel

Figure 3.21: Performance en fonction de la distance pour diverses configurations atmosphériques

- distance courte (*XX*)

Les conditions d'évaluation sont les suivantes:

- configuration RSC/RSS facile (configuration 2)
- σ_{atmo} bon (valeur de *XX*)
- toutes distances

Les performances sont présentées figure 3.22. La figure 3.23 présente deux véhicules, toujours les mêmes, vus à différentes distances. Quand la distance augmente et que la résolution diminue, ces véhicules sont de moins en moins discriminables. On comprend alors pourquoi la performance chute autant avec la distance.

3.7.10 Conclusion de l'étude des paramètres opérationnels

Dans les rangs de valeurs définis par les scénarios A,B,C et D, l'algorithme proposé est généralement stable en situation de déja-vu et de jamais-vu. En situation de déjà-vu, les données d'apprentissage et de test sont générées par le même phénomène. L'algorithme est très stable, sauf en cas d'occultations où il est un peu moins stable. En situation de jamais-vu, les données d'apprentissage sont faciles et les données de test sont plus difficiles. Dans ce cas, l'algorithme est très stable, sauf en cas d'occultations ou de RSC faible combiné à des cibles extraites avec imprécision. Pour palier ces deux problèmes, nous proposons de simuler des occultations lors de l'apprentissage pour y devenir plus robuste, et d'utiliser le masque des cibles lors de l'apprentissage et de la prédiction pour éliminer les informations provenant du fond.

Nous avons aussi vu que la performance diminue avec le nombre de classes à prédire, mais nous ne disposons pas des données suffisantes pour prédire le comportement de l'algorithme avec un nombre élevé de classes.

Notons qu'il est dans la nature des algorithmes d'apprentissage de mieux se comporter dans une situation de déjà-vu que de jamais-vu, car ceux-ci sont basés sur une capacité à

Figure ou tableau confidentiel

Figure 3.22: Performance en fonction de la distance

Figure ou tableau confidentiel

Figure 3.23: Deux véhicules (toujours les mêmes) vus à différentes distances, et la performance correspondante. On constate que la performance chute avec la perte de résolution: les véhicules sont de moins en moins discriminables.

généraliser, i.e. interpoler[17]. Si une forme inconnue ne peut être produite par interpolation de formes connues, l'algorithme ne dispose pas des éléments suffisants pour faire une prédiction correcte.

3.8 Informations pratiques

Cette section donne des informations pratiques sur l'échantillonnage, les temps de calculs et donne des matrices de confusions de cas intéressants.

3.8.1 Échantillonnage

Considérons une ROI de taille 100x200. Dans cette section nous calculons le nombre de positions échantillonnées lors de la détection de mots visuels dans l'image.

Dans un premier temps, considérons la configuration paramétrique ci-dessous:

- $D = 4000$ mots visuels
- $off = 0.2$
- $H_{max} = 57$ pixels
- $H_{min} = 20$ pixels
- $ds = 0.81$
- $ns = 6$
- $w_{std} = 12$ pixels

La pyramide d'échelles est composée d'images de taille: 57x114 , 46x92, 37x75, 30x61, 24x49 et 20x40. L'offset en x et en y est de 0.2*12=2.4 pixels, que l'on arrondi a 2 pixels. Cela signifie qu'un pixel sur 4 ($2 * 2$) est considéré à chaque échelle. Le nombre de positions échantillonnées est donc: (57-12+1)/2*(114-12+1)/2 + (46-12+1)/2*(92-12+1)/2 + (37-12+1)/2*(75-12+1)/2 + (30-12+1)/2*(61-12+1)/2 + (24-12+1)/2*(49-12+1)/2 + (20-12+1)/2*(40-12+1)/2, soit 2735. Comme le vocabulaire visuel comporte 4000 éléments, il faut calculer 2735* 4000=10.94 millions de NCC. Chaque NCC se calcule en 12x12=144

[17] au sens commun du terme, et non au sens mathématique

opérations (une opération consiste à calculer deux additions et une multiplication)[18] . Le calcul d'un descripteur se fait donc en 1.6 milliard d'opérations.

Considérons ensuite une configuration paramétrique plus légère:

- $D = 1000$ mots visuels
- $off = 0.5$
- $H_{max} = 50$ pixels
- $H_{min} = 50$ pixels
- $ns = 1$
- $w_{std} = 10$ pixels

Dans cette configuration, il y a (50-10+1)/(10*0.5)*(100-10+1)/(10*0.5)= 150 positions échantillonnées, et donc 150*1000*(10*10)= 15 millions d'opérations à effectuer. Soit 100 fois mois que dans la configuration précédente.

Notons que ce nombre d'opérations est donné pour des images ne contenant pas de régions uniformes. En effet, la NCC entre une région uniforme et une région quelconque est définie comme valant 0.5, la NCC est donc connue sans effectuer d'opérations.

3.8.2 Temps de calcul

Nous donnons ci-dessous le temps de calcul des différentes étapes de l'algorithme, sur un PC i686 Intel Pentium 4 CPU 3.40GHz, sur lequel tourne une distribution Linux Mandriva de noyau 2.6.11-6.

Calcul d'une description

La configuration paramétrique utilisée dans cette section est confidentielle. Le temps de calcul des 19595 descriptions est de 36.6 heures avec un code C++ optimisé, ce qui correspond à un temps de calcul de l'ordre de 7 s par image.

Apprentissage d'un SVM linéaire

Si le vocabulaire visuel est de taille 4000 et qu'il y a 4 seuils NCC d'activation, les descriptions ont une taille de 16000. Si on considère environ 3000 exemples d'apprentissage (scénarios A,B,C) et 7 classes d'objet, le temps d'apprentissage du SVM linéaire est de l'ordre de 5 minutes avec un code C++ optimisé.

[18] pour chaque pixel (i,j) d'un mot visuel $\underline{P}$ normalisé en moyenne et variance, et d'une région locale échantillonnée Q de moyenne $\overline{Q}$, $NCC \leftarrow NCC + P_{i,j} * (Q_{i,j} - \overline{Q})$.

Prédiction d'un label avec SVM linéaire

La prédiction d'une classe parmi 7 pour une description de taille 16000 par un SVM linéaire est de l'ordre de 70 ms avec un code C++ non optimisé.

3.8.3 Matrices de confusion

Dans cette section nous donnons quelques exemples de matrices de confusion. Nous considérons différents cas de figure. Ils ont en commun d'utiliser les images des scénarios A,B et C pour l'apprentissage, et les images du scénario D pour le test. La somme des éléments de chaque ligne des matrices de confusion n'est pas de 100 en raison des arrondis.

Configuration optimale

Nous considérons la configuration paramétrique ayant donné les meilleurs résultats: celle-ci est confidentielle.

La matrice de confusion obtenue est présentée tableau 3.8.

Influence du nombre d'exemples d'apprentissage

Nous observons maintenant les matrices de confusion (tableau 3.9) lors d'un apprentissage avec 10, 25 et 150 exemples d'apprentissage par classe. Les paramètres de détection sont constants pour les trois expérimentations. Les confusions diminuent rapidement avec l'augmentation du nombre d'exemples par classe.

Influence du nombre de classes

Observons maintenant les matrices de confusion quand le nombre de classes cibles est 2, 3 ou 7 (tableau 3.10). Les paramètres de détection sont constants pour les trois expérimentations. Nous rappelons ci-dessous la définition des groupes:

- les 7 classes de véhicules présents dans la base de donnée CEP
- les 3 groupes de classes définis par le CEP:
  - I : chars lourds
  - II : véhicules blindés de transport de troupes, VBTT
  - III: véhicules blindés de combat d'infanterie = VBCI

Figure ou tableau confidentiel

Table 3.8: Matrice de confusion en configuration optimale

Figure ou tableau confidentiel

Table 3.9: Variation du nombre d'images d'apprentissage par classe. De haut en bas, matrices de confusions obtenues avec 10, 25 et 150 images par classe.

- les 2 groupes de classes définis par Bertin:
  - I: *XX* ... *XX*
  - II: *XX* ... *XX*

On observe que comme il a été dit section 3.7.5, la performance globale augmente quand le nombre de classes diminue, en raison de la disparition des confusions au sein des sous ensembles de classes définis.

3.9 Conclusion

3.9.1 Étude de l'algorithme

Nous avons présenté de manière très détaillée l'algorithme de reconnaissance et identification spécifique à la problématique infra-rouge. Nous avons présenté les différents paramètres de cet algorithme, et avons constaté qu'il est aisé de définir les valeurs optimales de ces paramètres:

- La taille du vocabulaire visuel (1500 ici) peut être déterminés lors de l'apprentissage.
- Le nombre d'exemples d'apprentissage par classe (128 ici) peut aussi être déterminés lors de l'apprentissage.
- Les meilleurs paramètres d'échantillonnage sont ceux qui offrent l'échantillonnage le plus dense, l'intérêt d'échantillonner de manière moins dense étant uniquement d'accélérer le temps de calcul des descriptions.
- La taille relative des régions locales par rapport à la hauteur des fenêtres doit être de *XX* à *XX* , ce qui est assez intuitif: cela correspond à peu près à la taille d'une roue dans une région d'intérêt.

Figure ou tableau confidentiel

Table 3.10: Matrices de confusions avec 7 classes, ou avec des regroupements en 3 ou 2 groupes de classes

- L'utilisation de quatre seuils NCC d'activation est plus efficace que l'utilisation d'un seul seuil. Les quatre seuils offrent une bonne discrétisation des NCC continues entre les images et les mots visuels.

- Les SVMs RBF sont plus performants que les SVM linéaires, comme l'indique la littérature.

L'étude des paramètres algorithmiques s'est faite en utilisant les scénarios A,B,C et D de la base d'images hybrides du CEP. Les trois premiers ont servi à l'apprentissage et le dernier à l'évaluation. C'est une tâche difficile car le scénario D contient des occultations, ce qui n'est jamais vu lors de l'apprentissage. De plus, de nombreux paramètres fixes dans les scénarios A,B et C varient en même temps dans la base D.

La configuration présentée ci-dessus atteint une performance de *XX* . D'une manière générale, en fonction des différents réglages des paramètres, la performance varie entre *XX* et *XX* . Rappelons qu'une prédiction aléatoire avec 7 classes obtient une performance de 14%.

On peut donc considérer qu'il est aisé de régler les paramètres présentés ci-dessus. Il existe cependant des *paramètres critiques,* ce sont les paramètres du classifieur. Cependant, ceux-ci ne sont pas réglés manuellement, mais automatiquement à partir des données d'apprentissage. D'où la **nécessité** d'avoir, lors de l'apprentissage, des exemples typiques des situations à décrire. Sans cela, les paramètres réglés automatiquement ne seraient pas adaptés aux données dont la classe doit être prédite.

3.9.2 Étude des paramètres opérationnels

Nous avons aussi effectué une étude sur les paramètres opérationnels. Dans le cadre des images hybrides que nous avons traitées, il en ressort que les deux difficultés sont les *occultations* et la présence de *fond texturé* en cas d'extraction de ROI imprécise. Nous avons proposé deux solutions pour palier ces problèmes: simuler des occultations lors de l'apprentissage, et utiliser un masque de détection précis lors de l'apprentissage et de la prédiction.

FOURTH CHAPTER

Classification multi-classes: compromis temps-performance

4.1 Résumé du chapitre

Dans ce chapitre, nous proposons un nouvel algorithme de classification multi-classes d'images. Nous considérons toujours une représentation d'images par sac-de-mots, comme dans les chapitres précédents, et nous proposons un algorithme de classification qui optimise le compromis performance / temps d'execution. Nos contributions sont (a) une transformation de données haut-niveau basée sur des critères d'information mutuelle et (b) un algorithme de sélection de primitives adaptées à un classifieur SVM hiérarchique.

Notre approche a été évaluée sur des données réelles issues de caméras infra-rouge (classification de véhicules), et sur des catégories d'objets en imagerie visible. Sur la base infra-rouge, avec un facteur d'accélération des calculs de 100, le classifieur SVM hiérarchique que nous proposons obtient une performance 12% supérieure au classifieur standard SVM un-contre-un.

4.2 Introduction

Dans les chapitres précédents, nous avons étudié la représentation d'images par sac-de-mots, et avons déterminé les briques algorithmiques et les paramètres qui donnent le meilleur taux de réussite. Dans ce chapitre, nous ne nous intéressons pas seulement aux performances mais au compromis entre performance et temps de calcul. Le cadre que nous considérons est celui de la classification multi-classes, ce qui était aussi le cas pour la majorité des expérimentations des chapitres précédents.

Nous proposons un algorithme de classification de véhicules plus rapide que les algorithmes actuels, et qui conserve une haute performance: c'est l'enjeu de ce chapitre. Notre intérêt principal est la reconnaissance de modèles de véhicules en imagerie infra-rouge, mais nous montrons que cette approche peut être étendue à des catégories plus génériques en lumière visible.

Comme nous l'avons mentionné dans l'état de l'art section 1.3, les méthodes performantes récentes fonctionnent à base de motifs (templates) ou à base de modèle d'apparence. En ce qui concerne les méthodes à base de motifs [57], la reconnaissance est basée sur une mesure de distance (une corrélation normée centrée par exemple) entre l'image requête et le motif à rechercher. Les méthodes basées sur des modèles d'apparence [2, 49, 16, 94, 83, 45, 26, 68, 98, 64, 70, 35, 86] apprennent plutôt la variabilité de l'apparence des véhicules d'une même catégorie à partir d'un ensemble d'images d'apprentissage. Les images sont alors représentées par un ensemble de parties locales ou par des descripteurs globaux. Les premiers présentent un certain nombre d'avantages sur les seconds: robustesse aux modifications d'angle de vue, d'illumination, de forme, et robustesse aux occultations.

Dans la famille des représentations locales d'images, une méthode très populaire est la représentation par sac-de-mots, que nous avons analysée en détails dans le chapitre 2. Cela consiste à représenter une image par l'histogramme d'occurrences de parties locales typiques, appelées mots visuels, dont l'ensemble forme le vocabulaire visuel [16, 70]. Les travaux de ce chapitre sont basés sur cette représentation.

La meilleure performance de classification est obtenue quand les mots visuels sont recherchés de manière exhaustive [45, 70][1], et que le vocabulaire visuel est très large [16, 70] (contient une grande quantité de mots visuels). Ces deux conditions rendent les calculs très longs. Ils peuvent être accélérés en cherchant les mots visuels à un nombre réduit de positions, ou avec un vocabulaire visuel plus petit. Pour considérer moins de positions, il suffit d'utiliser un détecteur de points d'intérêt, mais ignorer certaines régions de l'image diminue significativement la performance [70], d'autant plus quand les images sont de petites tailles et bruitées, comme le sont les images infra-rouge que nous traitons. Pour réduire la taille du vocabulaire visuel, il suffit de sélectionner les parties les plus utiles à la classification. De telles parties sont illustrées figure 4.1.

[1] toutes positions et toutes échelles, par opposition à des positions éparses retournées par un détecteur de points d'intérêt

Nous avons étudié différentes méthodes de sélection de primitives, et leurs avons toutes trouvées des limitations dans le cas d'une utilisation multi-classes. C'est pourquoi notre première contribution (section 4.6.1) est une méthode de sélection de primitives basée sur un classifieur hiérarchique multi-classes, qui sélectionne uniquement les primitives utiles aux différents niveaux de la hiérarchie. Notre seconde contribution est une nouvelle représentation d'images haut-niveau obtenue par transformation du descripteur sac-de-mots originel (section 4.6.2).

Nous présenterons rapidement des méthodes de représentation d'images par parties (section 4.3) car elles ont déjà été introduites plus haut (section 1.3.2 et chapitre 2). Puis nous présenterons diverses méthodes de classification multi-classes (section 4.3) et de sélection de primitives (section 4.5). Nous présenterons ensuite notre algorithme, qui fait appel à ces trois domaines, puis nos résultats expérimentaux.

4.3 Représentation d'objets par parties locales

Nous avons mentionné dans les précédents chapitres différentes méthodes de représentation d'image par parties. Afin que ce chapitre se suffise à lui même, nous rappelons ici quelques méthodes classiques pour effectuer les différentes étapes: calcul d'un vocabulaire visuel, détection des parties, combinaison des parties pour décrire l'image. La classification multi-classes est présentée section 4.4.

4.3.1 Calcul d'un vocabulaire visuel

Une image peut être considérée comme un très grand ensemble de parties locales superposées: ce sont les différentes régions de toutes tailles et à toutes positions qui la composent. Ces parties locales peuvent avoir une infinité d'apparences, mais on peut cependant définir

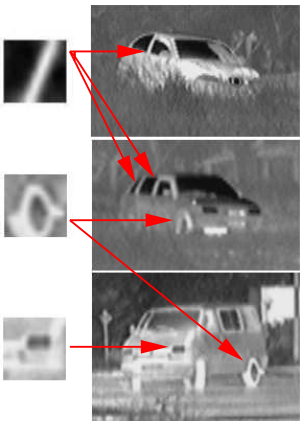


Figure 4.1: Mots visuels (issus du vocabulaire visuel) détectés dans les images. Sont-ils utiles pour différencier les différentes catégories d'objet?

un petit ensemble de parties locales typiques (des mots visuels) pour représenter chacune d'entre elles. Par exemple, dans la figure 4.1, les roues de la deuxième et de la troisième voiture sont représentées par le second mot visuel (à gauche). Cela signifie que l'on estime que les différences entre la partie locale (roue 1 ou roue 2) et le mot visuel (roue typique) sont faibles, alors on représente ces roues spécifiques par la roue générique du vocabulaire visuel.

Le vocabulaire est généralement obtenu par un algorithme de quantification. Les descripteurs locaux (niveaux de gris de régions de l'image par exemple) sont tout d'abord extraits à différentes orientations et échelles, puis ils sont regroupés par similarité. Chaque groupe est représenté par un élément caractéristique, l'ensemble de ces éléments caractéristiques forme le vocabulaire visuel.

Leug et Malik [50] utilisent une banque de filtres de convolution pour décrire des pixels échantillonnés selon une grille régulière, et le vocabulaire visuel est produit par l'algorithme des k-moyennes. Agarwal [2] et Leibe [49] utilisent des détecteurs de points d'intérêt pour déterminer des régions saillantes de l'image, qui sont ensuite décrites par les valeurs en niveau de gris de la région située autour du point d'intérêt. Ils construisent un vocabulaire visuel par clustering agglomératif. Pour la quantification, Willamowski [93] utilise l'algorithme des k moyennes, et Jurie [45] utilise une estimation de densité basée sur l'algorithme mean-shift.

4.3.2 Détection de parties locales

[2, 49, 93] utilisent des détecteurs de points d'intérêt pour déterminer les régions à considérer dans les images, et de ce fait considèrent un nombre réduit de positions dans l'image. La détection est rapide, et les calculs qui s'en suivent le sont aussi en raison du faible nombre de régions à considérer, mais d'un autre côté la performance est très faible quand la détection n'est pas fiable, ce qui est le cas quand les images sont de petite taille, bruitées, ou que les régions informatives ne peuvent être retournées par le détecteur de points d'intérêt choisi, ce qui est le cas pour nos images infrarouges. [86, 89, 45] suggèrent d'utiliser un échantillonnage (ou sélection) dense et multi-échelles de l'image, et de ce fait les mots de vocabulaire visuel sont détectés par corrélations par parcours systématique et multi-échelles. Cette étape nécessaire à une bonne performance [45] est très coûteuse en temps de calcul, c'est pourquoi les systèmes temps-réel doivent réduire la taille du vocabulaire visuel pour fonctionner plus rapidement (voir section 4.7.4).

4.3.3 Représentation d'images

Agarwal [2] et Leibe [49] utilisent des contraintes géométriques entre les différentes parties d'image pour modéliser les relations entre parties locales. Agarwal prend en compte toutes les relations deux-à-deux en quantifiant la distance et l'orientation. Leibe modélise la positon des parties par rapport au centre de l'objet. D'un autre côté, l'approche par sac-de-mots de Willamowski [93] ignore les contraintes géométriques entres les différentes parties

d'images. Les images sont tout simplement représentées par un histogramme qui compte les occurrences des mots visuels dans l'image. Cette représentation simple donne aujourd'hui les meilleurs résultats de l'état de l'art [45].

4.4 Classification multi-classes

Cette section présente des classifieurs multi-classes de l'état de l'art adaptés à une représentation d'images par sac-de-mots. On dispose donc d'un ensemble d'images d'apprentissage, à chacune est associé un label (numéro de catégorie compris entre 1 et C, avec C le nombre total de catégories) et un vecteur descripteur V dont la dimension est égale à la taille du vocabulaire visuel. Chaque dimension de V représente les occurrences du mot visuel correspondant dans l'image. Le classifieur multi-classes est une fonction qui prend en entrée le descripteur V d'une image quelconque et qui retourne le label prédit pour cette image.

Le classifieur le plus simple est le classifieur plus proche voisins. Il consiste à assigner à une donnée la classe majoritaire parmi les k exemples d'apprentissage les plus proches (distance euclidienne ou Mahanalobis ou autre), où k est un paramètre [15]. La performance de ce classifieur est très dépendante de la mesure de distance utilisée. De plus, il requiert de stocker en mémoire tous les exemples d'apprentissage, et le temps de prédiction est linéairement dépendant du nombre d'exemples d'apprentissage.

Les arbres de décision [10] sont des classifieurs multi-classes populaires. Un vecteur descripteur est classé en passant de la racine de l'arbre à une feuille, la valeur prédite étant le label associé à la feuille. Le chemin dans l'arbre est déterminé par l'évaluation de conditions simples sur les dimensions de V (comparaison entre une valeur et un seuil le plus souvent). Leur construction étant basée sur la probabilité jointe des différentes primitives, il est préférable de choisir d'autres classifieurs quand la dimension des données devient trop importante par rapport à la quantité d'images disponible.

Le classifieur multi-classes génératif le plus populaire en haute dimension est sans doute le classifieur Bayesien naïf [77], car il suppose que toutes les primitives sont indépendantes, ce qui évite d'estimer des probabilités jointes. Ce classifieur modélise la densité de probabilité d'un descripteur d'image V (vecteur d'occurrences de mots visuel) appartenant à la catégorie C_k par $P(V|C_k) = \prod_j P(V_j|C_k)$. $P(V_j = 0|C_k)$ et $P(V_j = 1|C_k)$ sont estimés lors d'une phase d'apprentissage, par comptage par exemple. Une image inconnue décrite par un descripteur d'image V est alors affectée à la classe la plus probable, qui est celle qui maximise $P(V|C_k)$ dans le cas où les classes sont équiprobables.

Dans les articles récents de l'état de l'art, le classifieur discriminatif le plus fréquemment utilisé est le SVM. Les SVMs ont été conçus pour la classification binaire [58, 88], et ont été ensuite étendus à la classification multi-classes, avec principalement les stratégies "un-contre-un" et "un-contre-tous".

Les classifieurs un-contre-tous entraînent un classifieur binaire par classe, cette classe contre toutes les autres: classe 1 contre toutes les autres, classe 2 contre toutes les autres, ..., classe C contre toutes les autres. La classe prédite est alors la classe associée au classifieur

dont la valeur de confiance est la plus élevée. Les classifieurs un-contre-tous présentent deux problèmes majeurs. Premièrement, les différents classifieurs ne sont pas optimisés en même temps, rien ne garantit donc que les valeurs issues des classifieurs sont directement comparables. Deuxièmement, ces classifieurs sont asymétriques, puisque si les classes sont équiprobables, les différents classifieurs appris ont des rapports d'$1/C$ entre les classes positives et les classes négatives, ce qui peut rendre l'apprentissage plus délicat.

Les classifieurs un-contre-un entraînent tous les classifieurs binaires possibles en considérant les classes deux-à-deux: classe 1 contre 2, 1 contre 3, ..., 1 contre C, 2 contre 3, ..., $C-1$ contre C. Cela requiert donc l'apprentissage de $C(C-1)/2$ classifieurs binaires. Lors de la prédiction, tous ces classifieurs binaires sont évalués et la classe prédite est celle qui remporte le plus de compétitions. Bien que le nombre de classifieurs binaires à apprendre soit plus élevé que dans le cas des classifieurs un-contre-tous, chaque classifieur traite moins de données (deux classes au lieu de C classes). De plus, les classifieurs un-contre-un ne souffrent pas des deux problèmes cités plus haut, ils sont donc en général préférés par la communauté.

Récemment, Rajan [74] a proposé une hiérarchie de classifieurs binaires, utilisant un arbre binaire de classifieurs binaires pour résoudre les problèmes de classification multi-classes (voir figure 4.2). Le noeud racine de l'arbre contient l'ensemble des classes considérées. Chaque noeud est alors récursivement séparé en deux ensembles de classes disjoints, et un classifieur est appris pour séparer ces deux ensembles. L'opération se poursuit jusqu'à l'obtention de feuilles ne contenant qu'une seule classe. Un exemple de classe inconnue est classé en parcourant les noeuds de la racine à une feuille: le classifieur associé au noeud courant prédit si l'exemple appartient aux classes du fils droit ou du fils gauche, puis l'exemple passe dans ce noeud fils, et poursuit itérativement un chemin jusqu'à une feuille. Le label prédit pour l'exemple de classe inconnue est celui associé à la feuille qu'il atteint.

Rajan construit cet arbre par algorithme des k-moyennes itératif. Pour cela, il calcule un descripteur pour chaque image d'apprentissage, puis il calcule la moyenne des descripteurs de chaque classe, qui sert de représentant à cette classe. Le noeud racine de la hiérarchie contient le représentant de toutes les classes. Puis ce noeud, comme tous les suivants, est itérativement partitionné en deux groupes: un algorithme de k-moyennes avec $k=2$ est appliqué sur ces descripteurs représentatifs, ce qui permet d'obtenir deux ensembles de classes. Dans le cas de la figure 4.2, les deux groupes issus du noeud racine contiennent les classes 1-2-4 et 3-5. A chaque fois que deux groupes de classes sont créés, on apprend un classifieur binaire qui les sépare. Le processus est répété avec les noeuds fils créés jusqu'à l'obtention de feuilles ne comportant qu'une seule classe.

4.5 Sélection de primitives

Le vocabulaire visuel est composé d'un ensemble de mots visuels. Notre intention est d'accélérer les calculs en réduisant la taille du vocabulaire visuel, en choisissant les éléments les plus *utiles* pour la classification. Avant sélection de primitives, une image est décrite par

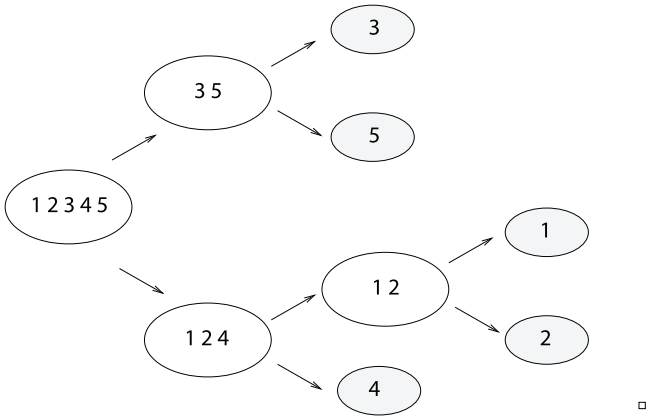


Figure 4.2: Hiérarchie de classifieurs binaires. Chaque noeud représente un ensemble de classes. Les classes sont itérativement séparées jusqu'à l'obtention de feuilles ne contenant qu'une seule classe.

un descripteur V de dimension égale à la taille du vocabulaire visuel. Pour obtenir un tel descripteur, il faut donc détecter tous les éléments du vocabulaire visuel dans l'image, i.e. comparer chaque région locale considérée à *tous* les mots du vocabulaire visuel, long processus si le vocabulaire est de grande taille. La sélection de primitives consiste à choisir les mots visuels les plus utiles à la classification, dans ce cas une image sera décrite par un vecteur V' de dimension inférieure à celle de V. Seuls les mots visuels correspondant aux dimensions de V' seront détectés dans l'image, ce qui rend les calculs plus rapides.

La sélection de primitives est différente de l'extraction de primitives. L'extraction de primitives crée un nouvel ensemble de primitives, chacune étant une fonction des primitives d'origine. Une méthode simple d'extraction de primitives consiste à faire une analyse en composantes principales, puis à garder les vecteurs propres correspondant aux valeurs propres les plus élevées, où se concentre la grande partie de la variance des données. Mais chaque nouvelle primitive étant une combinaison linéaire des primitives d'origine, cela nous obligerait à détecter l'ensemble des éléments du vocabulaire visuel pour calculer la nouvelle représentation. Cela ne présente aucun intérêt, puisque notre but est de détecter le moins de mots du vocabulaire visuel possible (gain de temps proportionnel au ratio des tailles des vocabulaires visuels).

La sélection de primitives [62, 75, 33, 78, 18, 52, 97, 32] est donc l'outil adapté à nos besoins. Elle consiste à choisir le sous ensemble le plus utile à une tâche, qui est la classification dans notre cas. C'est un domaine qui a été beaucoup étudié dans la classification de texte [97, 33, 78, 52, 62]. Nous détaillons ci-dessous les principales techniques de sélection de primitives.

Étant donné un vocabulaire visuel de n éléments, les algorithmes de sélection de primitives [44] sélectionnent les $n' < n$ primitives les plus utiles à la classification. La recherche du sous ensemble optimal est un problème NP-complet, car il requiert la comparaison de

tous les vocabulaires visuels de taille n'. En pratique cette approche n'est pas possible, et on utilise une recherche sous-optimale. L'algorithme de sélection de primitives ne donne pas la taille optimale n' du sous-ensemble, celle-ci peut être déterminée en étudiant la performance de classification sur un ensemble de validation.

La méthode de sélection de primitives dite "wrapper" construit le sous ensemble de taille n' itérativement. Le sous ensemble est initialement vide. Itérativement, le mot visuel qui augmente le plus la performance (taux de classification multi-classes par exemple) est ajouté au sous-ensemble, jusqu'à ce qu'il atteigne une taille de n' éléments. La section 4.7.4 montre que parmi les méthodes considérées celle-ci donne la meilleure performance. Son grand point faible est sa complexité algorithmique quadratique ($O(nn')$).

Une approche moins coûteuse fait l'hypothèse que toutes les primitives sont indépendantes. Une mesure d'utilité est alors calculée pour chaque mot visuel, et le sous-ensemble choisi est alors fait des n' mots visuels ayant le score le plus élevé. La complexité de ces algorithmes est linéaire ($O(n)$). De nombreuses mesures d'utilité ont été proposées dans le domaine de la classification de textes [62], nous mentionnons ci-dessous les mesures les plus populaires.

1. Sélection aléatoire, cela consiste à sélectionner uniformément aléatoirement n' primitives parmi n. Cette méthode n'optimise aucun critère, mais permet de mesurer la performance relative des autres méthodes.

2. Fréquence des termes, afin de favoriser les informations les plus disponibles. La fréquence d'un terme est le ratio entre le nombre de descripteurs dans lequel il apparaît et le nombre total de descripteurs.

3. L'information mutuelle [14][2], afin de mesurer l'utilité d'un mot de vocabulaire visuel pour une tâche de classification. Soient C une variable aléatoire discrète qui dénote une catégorie d'objets, et V_j une variable aléatoire binaire qui dénote la présence du mot visuel j dans une image. L'information mutuelle entre ces deux variables aléatoires est définie par

$$I(V_j,C) = \sum_{v_j=0,1}\sum_{c=1..C} p(C=c)Freq_{j,c}\log(Discr_{j,c}) \quad (4.1)$$

$$Freq_{j,c} = p(V_j=v_j|C=c) \quad (4.2)$$

$$Discr_{j,c} = \frac{p(V_j=v_j|C=c)}{\sum_c p(V_j=v_j|C=c)p(C=c)} \quad (4.3)$$

Cette mesure combine donc les statistiques de fréquences du mot visuel j ($Freq$) ainsi que son pouvoir discriminant ($Discr$). Si un mot est très fréquent mais peu discriminant (i.e. apparaît avec la même probabilité dans toutes les classes), alors il n'apporte pas d'information. De même, un mot peu fréquent mais très discriminant ne sera pas

[2] appelée information mutuelle dans [14] et gain en information dans [97]

utile, car bien que son apparition augmente énormément la confidence d'appartenance à une classe, cette apparition est trop rare pour être utile. Au contraire, le mot le plus utile est celui qui apparaît toujours dans les images de certaines catégories et jamais dans les autres.

4. Rapport des côtes (Odds Ratio), afin de sélectionner les primitives les plus discriminantes. Comme les informations de fréquence ne sont pas prises en compte, cette mesure peut donner un score élevé à une partie très discriminante mais très rare. Pour que de telles parties soient utiles, il faut qu'elles existent en grande quantité dans le vocabulaire visuel (pour assurer qu'une quantité minimum d'entre elles est détectée), ce qui ne favorise pas la création de vocabulaires visuels compacts. Le rapport des côtes est défini par:

$$OR = \frac{p(V_j = 1|C = 1)}{p(V_j = 0|C = 1)} \frac{p(V_j = 0|C = 2)}{p(V_j = 1|C = 2)} \tag{4.4}$$

5. Les coefficients de la normale à l'hyperplan séparateur d'un SVM linéaire [75], afin de mettre l'accent sur les parties discriminantes. Pour cela, on entraîne un classifieur SVM linéaire sur les descripteurs des deux classes à discriminer, ce qui produit un vecteur $\omega = [\omega_1 ... \omega_n]$, qui décrit la normale à l'hyperplan séparateur de ces deux classes. La prédiction du SVM pour un descripteur $V = [V_1 ... V_n]$ de catégorie inconnue est: $svm(V) = \omega.V + b$, où b est une constante. Si on considère que les valeurs des V_j (pour différents j) sont du même ordre de grandeur, alors plus la valeur absolue de ω_j est élevée, plus la primitive V_j influence la décision de classification, ce qui motive l'utilisation de $abs(\omega_j)$ comme mesure d'utilité du mot visuel j.

Ces méthodes de sélection de primitives sont classiques, et la section suivante indique comment combiner au mieux ces méthodes de sélection de primitives et les hiérarchies de classifieurs binaires.

4.6 Méthode proposée

Cette section présente l'algorithme proposé pour la classification de véhicules en imagerie infra-rouge[3]. Nous proposons une méthode de sélection de primitives (section 4.6.1) et une nouvelle représentation de données haut niveau (section 4.6.2) qui s'intègrent bien avec les hiérarchies de classifieurs binaires.

4.6.1 Sélection de primitives pour hiérarchie de classifieurs binaires

Les classifieurs bayésien naïfs, SVM un-contre-tous et SVM un-contre-un requièrent chacun l'évaluation de C, C et $C(C-1)/2$ classifieurs. Si on applique une méthode de sélection

[3] valable aussi comme nous le montrerons pour des catégories d'objets génériques en imagerie visible

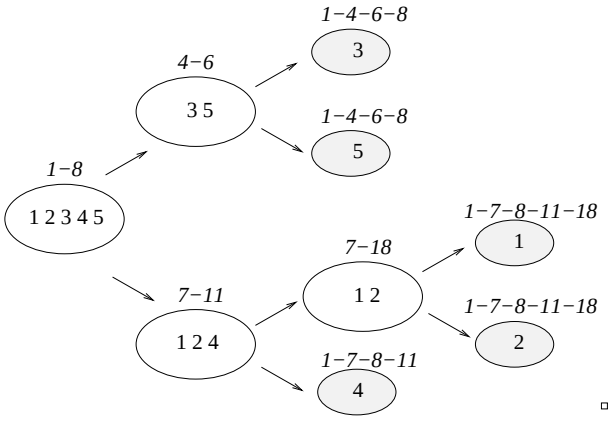


Figure 4.3: Sélection de primitives. Pour chaque noeud interne, les deux primitives utilisées par le classifieur associé au noeud sont indiquées en haut des noeuds. L'ensemble des primitives utilisées pour prédire les classes associées aux feuilles sont indiquées au dessus des feuilles.

de primitives multi-classes décrite dans la section 4.5, on obtient un sous-ensemble de primitives. Ces primitives sont les plus utiles au classifieur multi-classes global, mais ce ne sont pas les primitives les plus utiles à *chacun* des classifieurs binaires. Une primitive peut par exemple être très utile pour différencier deux classes, mais inutiles pour toutes les autres classifications binaires. Comme les classifieurs des hiérarchies de classifieurs binaires sont évalués séquentiellement, ils ne souffrent pas de cet inconvénient. Le principe est illustré figure 4.3, qui présente une hiérarchie de classifieurs binaires. La racine de l'arbre contient les 5 classes 1,2,3,4 et 5. Ces classes sont successivement divisées jusqu'à l'obtention de feuilles qui ne contiennent qu'une classe. Au dessus des noeuds internes, on indique les primitives sélectionnées et utilisées pour la classification binaire. Au dessus des feuilles, on indique les primitives utilisées par l'ensemble des classifieurs binaires qui ont été évalués pour atteindre cette feuille.

Supposons qu'une image doit être analysée par la hiérarchie de classifieurs binaires de la figure 4.3. Elle passe donc dans la racine de l'arbre, les mots visuels 1 et 8 sont détectés dans l'image. Le classifieur associé au noeud racine, et qui n'utilise que les primitives 1 et 8, prédit si l'image appartient aux classes 1,2 et 4 ou bien 3 et 5. Si c'est le premier cas, l'image passe dans le noeud "1,2,4". De ce fait, les primitives 4 et 6, associées uniquement à l'autre noeud "3,5" ne seront jamais détectées: elles sont inutiles. Les primitives 7 et 11 sont alors détectées dans l'image, et en fonction du résultat du classifieur associé au noeud "1,2,4", l'image passe dans le noeud "1,2" ou "4". Si elle atteint le noeud "4" qui est une feuille, la classe prédite est la classe 4. Sinon le processus se poursuit dans l'autre noeud: les primitives 7 et 18 doivent être détectées. La primitive 7 ayant déjà été détectée, seule la primitive 18 doit encore être détectée. Le classifieur associé au noeud "1,2" envoye l'image dans les feuilles "1" ou "2", ce qui détermine la classe prédite. Ce procédé est efficace, car 7 mots visuels différents sont utilisés pour la classification (1,4,6,7,8,11,18),

mais en moyenne une classe est prédite en utilisant $(4+4+5+5+4)/5 = 4.4$ mots visuels. Comme la détection des mots visuels est le goulot d'étranglement de l'algorithme, le gain en temps de calcul est très significatif.

C'est ce gain moyen en temps de calcul qui nous motive pour intégrer les différentes méthodes de sélections de primitives présentées section 4.5 aux hiérarchies de classifieurs binaires. La section courante sert d'illustration à notre philosophie, la démarche détaillée se trouve en section 4.6.3.

4.6.2 Transformation de données

Les approches par sac-de-mots représentent une image I_i par un vecteur V^i, où V^i_j représente l'occurrence de la primitive j, c'est à dire le nombre de fois que le mot visuel numéro j a été détecté dans l'image. Cette information brute ne peut être utilisée efficacement par les classifieurs SVM linéaires, outils qui obtiennent aujourd'hui les meilleures performances de l'état de l'art. En effet, l'apprentissage d'un SVM linéaire requiert le calcul d'une distance euclidienne entre deux vecteurs supports V^l et V^m: $d(V^l,V^m)^2 = \sum_{k=1..n}(V^l_k - V^m_k)^2$. Si la magnitude de la première dimension est très supérieure aux autres dimensions (mot visuel très fréquent par exemple), cette distance devient $d(V^l,V^m)^2 \simeq (V^l_1 - V^m_1)^2$, et l'information portée par les autres dimensions disparaît.

Différentes méthodes de normalisation des vecteurs descripteurs ont été proposées, nous en énumérons ci-dessous.

1. Données brutes: les vecteurs ne sont pas modifiés, les classifieurs utilisent les occurrences des mots visuels, i.e. V^i_j est le nombre de fois que le mot visuel j apparaît dans l'image i.

2. Transformations linéaires, par primitives. Chaque dimension du vecteur descripteur subit une transformation linéaire de telle sorte que les valeurs minimum et maximum parmi toutes les images d'apprentissage soient 0 et 1. Cela permet de donner le même ordre de grandeur à toutes les dimensions.

3. Transformation linéaire, par image. Chaque descripteur d'image est normalisé indépendamment: les éléments sont multipliés par une constante pour que leur somme fasse 1. De ce fait, V^i_j représente la probabilité qu'un mot visuel échantillonné dans l'image i soit le mot visuel j.

4. Binarisation simple. Si le mot visuel j est détecté au moins une fois dans l'image i, alors $V^i_j = 1$ sinon $V^i_j = 0$. L'information utilisée par le classifieur est alors la présence ou l'absence d'un mot visuel dans l'image.

Nous proposons une autre transformation, qui permet au classifieur SVM linéaire de tirer aisément parti d'une information de plus haut niveau. Cette transformation consiste à binariser chaque dimension du vecteur descripteur en utilisant un seuil de binarisation adaptatif.

Ce seuil est choisi parmi une liste de candidats formant une suite géométrique (1,2,4,8,...) de manière à maximiser le taux d'information mutuelle entre la primitive binarisée et les catégories d'objet. En effet, imaginons une situation où un mot visuel j apparaît dans toutes les images d'une base de données. Alors sa présence n'est pas informative, et la binarisation simple en fait une primitive inutile. Mais si ce mot apparaît souvent dans les images d'une catégorie (nombreuses détections par image, soit V_j^i élevé) et rarement dans les images d'une autre catégorie (peu de détections par image, soit V_j^i faible), alors "une quantité minimum de détections" devient une primitive très informative, puisqu'elle permet de différencier ces deux classes.

La binarisation simple supprime cette information de quantité de détections *minimum*. Si aucune transformation n'est appliquée, cette information est contenue, mais seulement de manière implicite, et les classifieurs SVM linéaires ne peuvent en tirer parti facilement. Au contraire, la binarisation adaptative que nous proposons contient cette information explicitement: elle exprime la présence *significative* d'un mot visuel dans une image.

Le seuil de binarisation de la variable aléatoire V_j comptant les occurrences du mot visuel j est donc:

$$\arg max_{k \in \{1,2,4,8,16,...\}} I(V_j \geq k , C) \tag{4.5}$$

où $I()$ mesure l'information mutuelle de deux variables aléatoires, et C est la variable aléatoire qui représente la classe d'une image.

4.6.3 Algorithme de classification proposé

Dans cette section, nous rassemblons les différents éléments que nous avons introduits et motivés plus haut, et présentons précisément les différentes étapes nécessaires à l'apprentissage et à l'évaluation de notre modèle de classification multi-classes.

Apprentissage du modèle

L'apprentissage du modèle consiste à (a) apprendre un vocabulaire visuel, (b) décrire les images d'apprentissage à l'aide de ce vocabulaire pour obtenir des vecteurs descripteurs, (c) transformer ces descriptions pour obtenir une représentation plus efficace, (d) apprendre une hiérarchie de classes, (e) effectuer une sélection de primitives pour chaque noeud interne de cette hiérarchie et (f) apprendre les classifieurs binaires des noeuds internes. Ces étapes sont détaillées ci-dessous.

Apprentissage du vocabulaire visuel. Le vocabulaire est construit par l'algorithme proposé par Jurie [45]. Il est démontré [45] que cet algorithme produit une quantification plus adaptée à la classification par sac-de-mots que l'algorithme des k-moyennes, car contrairement à l'algorithme des k-moyennes l'algorithme proposé par Jurie est basé sur mean-shift et produit des clusters de rayon constant, ce qui évite de créer des clusters attracteurs qui contiennent une trop grande variété de descripteurs.

Le vocabulaire est construit à partir de régions locales décrites par leurs niveaux de gris, échantillonnées de manière très dense à toutes positions et toutes échelles dans les images d'apprentissage. L'algorithme de Jurie recherche les concentrations de descripteurs dans l'espace des niveaux de gris grâce à l'algorithme mean-shift, et utilise une corrélation croisée normée centrée (NCC) pour comparer les différents descripteurs. Cela permet d'obtenir un vocabulaire de taille n.

Description des images d'apprentissage. Chaque image est représentée par un vecteur de taille n, chaque dimension du vecteur encodant le nombre de fois que le mot visuel correspondant est détecté dans l'image. De même que pour la création du vocabulaire, nous considérons un échantillonnage multi-échelle très dense des images: nous calculons une pyramide d'échelles d'un pas de 0.91, et toutes les régions de taille 10x10 sont considérées à chaque niveau de la pyramide. Si la corrélation croisée normée centrée (NCC) entre un mot visuel et une région locale est supérieure à un seuil (0.8 dans notre cas), alors le mot visuel est détecté dans cette image, et la dimension correspondante dans le vecteur descripteur est incrémentée.

Transformation des vecteurs descripteurs. Nous transformons les vecteurs descripteurs d'image à l'aide de la transformation que nous avons proposée section 4.6.2: la binarisation adaptative optimisée par taux d'information mutuelle. Nos expérimentations montrent (section 4.7.6) que cette transformation surpasse toutes les autres transformations présentées dans ce chapitre.

Apprentissage d'une hiérarchie de classes. L'algorithme de Rajan [74] présenté à la fin de la section 4.4 est utilisé pour apprendre une hiérarchie de classes, c'est à dire que l'algorithme des k-moyennes itératif ($k = 2$) est utilisé sur la moyenne des descripteurs de chaque classe pour déterminer la hiérarchie des classes.

Sélection de primitives. Nous disposons maintenant d'une hiérarchie de classes similaire à celle présentée figure 4.2. Nous effectuons ensuite une sélection de primitives dans chaque noeud interne, pour obtenir la liste des primitives les plus utiles à la séparation des groupes de classes associés à chaque noeud interne (algorithme détaillé section 4.4). Nous conservons dans chaque noeud interne les 40 primitives les plus utiles. Nos expérimentations montrent (section 4.7.4) que la sélection de primitives par taux d'information mutuelle est la plus avantageuse.

Apprentissage des classifieurs binaires. Dans chaque noeud interne, un classifieur binaire est appris pour séparer les groupes de classes des noeuds fils. Ce classifieur n'utilise que les primitives sélectionnées pour ce noeud (voir figure 4.3). Nous utilisons un SVM linéaire.

Évaluation du modèle

Le modèle appris comme indiqué ci-dessous est utilisé pour prédire la catégories d'images jamais vues lors de l'apprentissage. Pour cela, l'image parcourt la hiérarchie de classifieurs binaires de la racine à une feuille, comme indiqué section 4.4. Les différentes étapes sont détaillées ci-dessous, voir figure 4.3 pour une illustration.

1. L'image passe par la racine de l'arbre: la racine de l'arbre devient de noeud courant.
2. Les mots visuels associés au noeud courant sont recherchés dans l'image, et s'ils sont détectés viennent mettre à jour un vecteur descripteur (de taille le nombre de mots visuels associés au noeud).
3. Ce vecteur est transformé avec la même transformation qui a été utilisée lors de l'apprentissage
4. Le classifieur associé au noeud classifie ce vecteur descripteur transformé et prédit le groupe de classes auquel appartient l'image
5. L'image passe par le noeud fils correspondant au groupe de classes prédit
6. On itère les étapes 2 à 5 jusqu'à ce qu'une feuille soit atteinte, la classe prédite est alors la classe associée à cette feuille.

4.7 Expérimentations

Dans la section précédente, nous avons présenté deux contributions pour améliorer la classification par sac-de-mots: une méthode de sélection de primitives adaptée aux hiérarchies de classifieurs binaires, et une binarisation adaptative des descripteurs d'images. Nous allons montrer l'intérêt de ces méthodes sur une base d'images de voitures acquises par caméra infra-rouge. Pour démontrer que ces résultats ne sont pas spécifiques à l'imagerie infra-rouge, nous présenterons aussi des résultats sur une base de donnée en imagerie visible.

4.7.1 Bases de données

Cette section décrit les bases de données utilisées pour mesurer les performances de la méthode proposée.

La première base d'images est produite à partir d'une caméra infra-rouge. Les véhicules sont acquis sur fonds naturels chargés (voir figure 4.4). Les véhicules appartiennent aux catégories suivantes: fourgon, Citroen AX, Renault Scenic, Seat Ibiza. Les différentes catégories sont illustrées figure 4.5. Les difficultés de cette base sont illustrées dans différentes figures. Cette base contient des véhicules observés à différentes orientations (figure 4.5), différents niveaux d'occultation (figure 4.6), différentes échelles (figure 4.7) et différentes conditions thermiques (figure 4.8). L'objet d'intérêt occupe la majeur partie de l'image de

Figure 4.4: Images de la base Bertin acquises par une caméra infra-rouge.

manière à simuler les régions d'intérêt issues d'un détecteur de mouvement, voir figure 4.5. La base contient 250 images de chaque véhicule (soit 1000 images de véhicules) et 1500

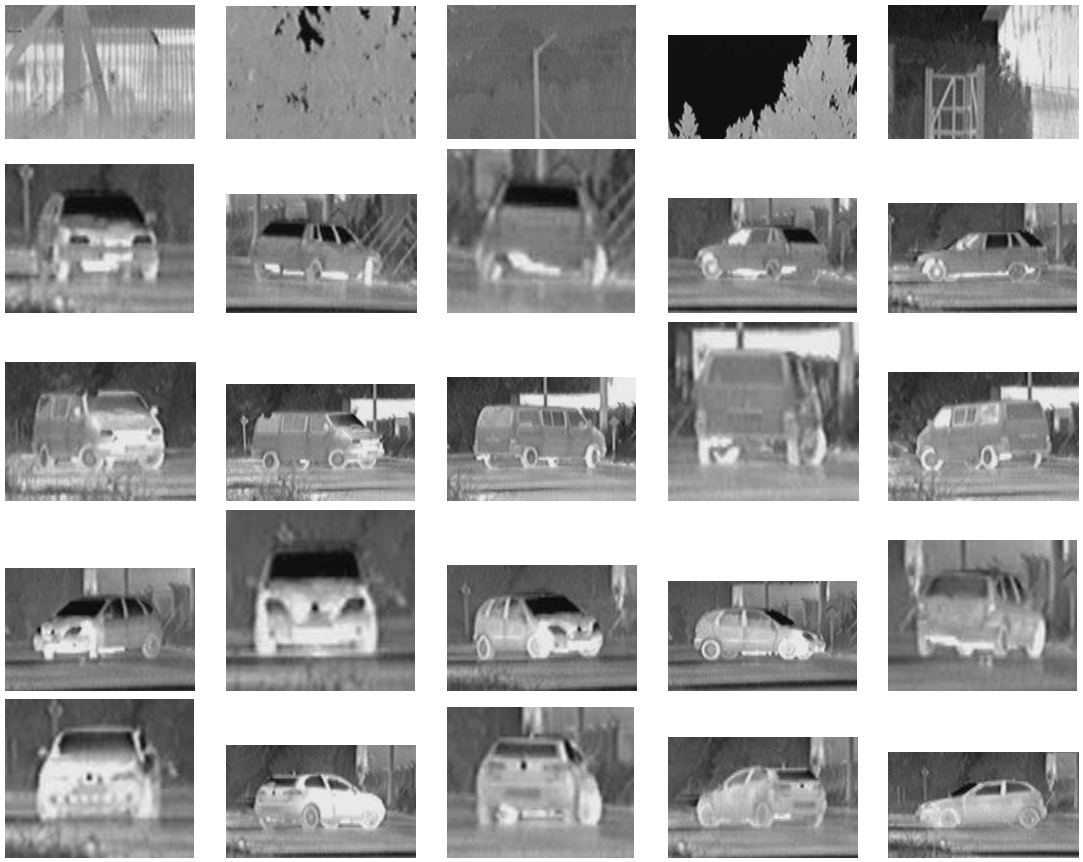


Figure 4.5: Images provenant de la base de données Bertin, acquises par caméra infra-rouge. Ligne 1: fond, ligne 2: Citroen AX, ligne 3: fourgon, ligne 4: Renault Scenic, ligne 5: Seat Ibiza. Pour chaque véhicule, les images présentées proviennent de la même séquence vidéo.

Figure 4.6: L'une des difficultés de la reconnaissance est l'occultation des véhicules. Ici le véhicule est une Citroen AX.

images de fond. L'algorithme de classification doit donc discriminer parmi 5 classes: 4 véhicules et le fond. On observe sur la figure 4.5 qu'il peut y avoir une grande similarité inter-classes, ce qui est une difficulté supplémentaire.

La seconde base de données contient des objets en lumière visible: 400 images de fond et 50 images pour chacune de ces sept catégories: cheval, visage, moto, voiture, vélo, livre, téléphone, voir figure 4.9. Toutes ces images sont issues de la base de données Lava-Xerox7.

Figure 4.7: En infra-rouge, l'éloignement d'un véhicule se traduit par un changement de taille apparente, mais aussi par un changement de résolution et une homogénéisation du signal reçu par la caméra. Les deux lignes montrent le même véhicule, vue arrière et avant. Colonne 1: véhicule vu à 100 mètres. Colonne 2: véhicule vu à 600 mètres. Colonne 3: agrandissement de la colonne 2. Par exemple la hauteur de l'image (telle qu'elle est acquise par la caméra) ligne 1 colonne 1 est de 71 pixels, celle de la ligne 1 colonne 2 est de 25 pixels.

Figure 4.8: En infra-rouge, les modifications thermiques influencent grandement l'apparence des objets. A gauche: AX moteur froid. A droite: AX moteur chaud.

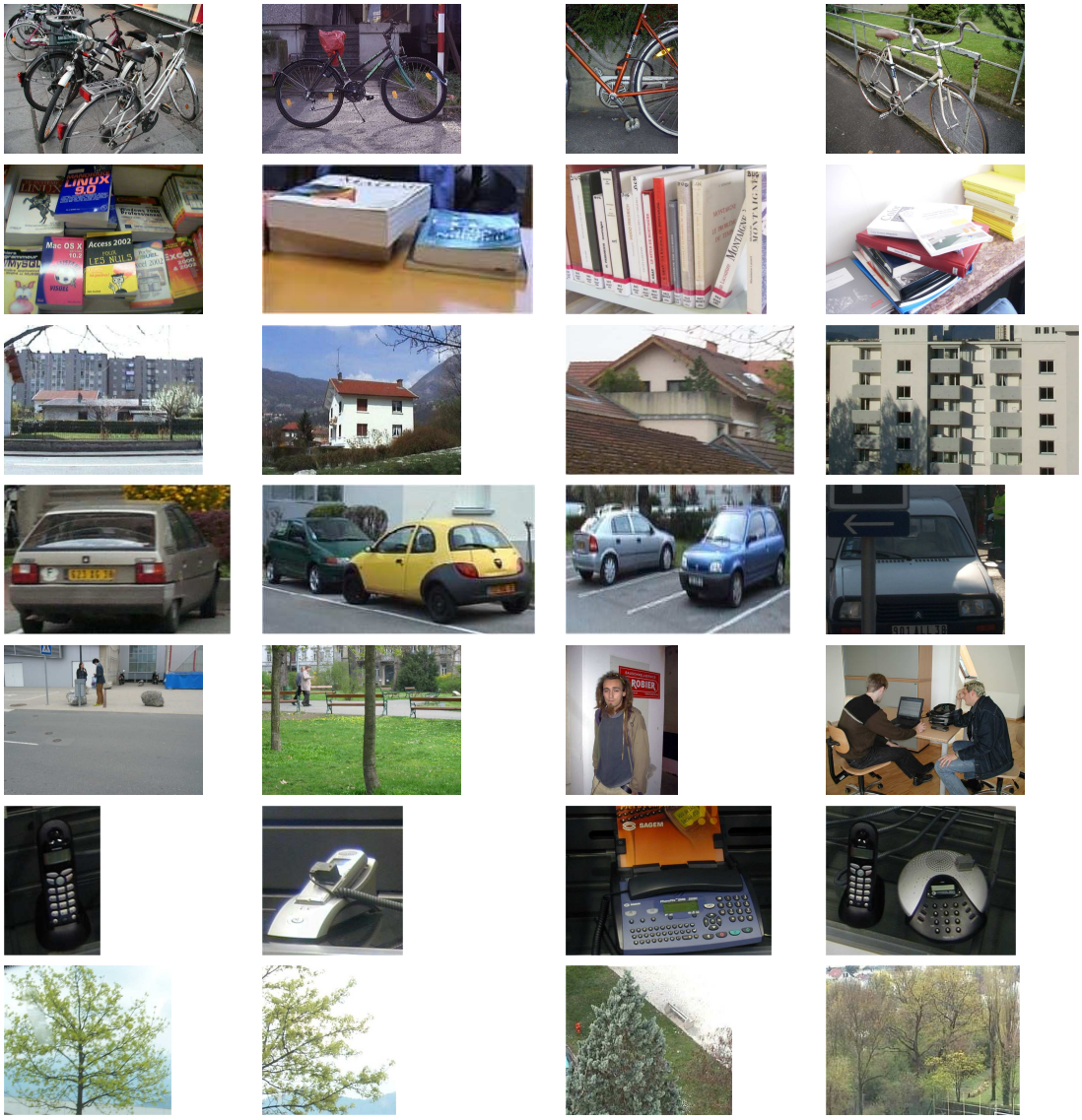


Figure 4.9: Images représentatives de la base Xerox7. De la première à la dernière ligne: vélos, livres, bâtiments, voitures, personnes, téléphones, arbres.

4.7.2 Mesure de performance

La performance est mesurée par validation croisée en 10 paquets. La performance est la moyenne du taux de reconnaissance par classe. Pour la base infra-rouge, nous effectuons aussi une mesure de performance en utilisant des images de véhicules acquises sur route pour l'apprentissage et des images dans un contexte différent (route aux conditions thermiques différentes, champ) pour l'évaluation. Cela permet de simuler la performance d'un

système réel en situation de jamais-vu, les détails et les performances de cette expérimentation spécifique sont présentés section 4.7.7.

Lors de la classification multi-classes avec une combinaison de classifieurs (SVM un-contre-un, un-contre-tous, hiérarchie de classifieurs binaires,...) il n'est pas possible de construire une courbe ROC ou Precision-Rappel et de prendre le taux d'erreur égale pour mesurer la performance. En effet, cette courbe est généralement obtenue en faisant varier le biais d'un classifieur, hors on dispose de plusieurs classifieurs. Il faut donc trouver une autre mesure de performance. Une mesure classique est la matrice de confusion. La valeur figurant ligne i et colonne j indique le nombre d'exemples de la classe numéro i qui ont été affectés à la classe numéro j. Cette matrice permet d'observer les différentes confusions, mais elle ne fournit pas un nombre réel pour mesurer la performance du classifieur, et donc de comparer plusieurs modèles.

Une telle mesure de performance peut être obtenue en comptant le nombre d'exemples bien classés: il s'agit de la trace de la matrice de confusion. Cependant, si une classe est majoritaire, les classifieurs qui favorisent cette classe sont favorisés. Comme la classe fond est majoritaire dans le cas de la base infra-rouge, cette mesure de performance serait biaisée. Nous utilisons donc une autre mesure de performance: la moyenne du taux de reconnaissance par classe, qui ne souffre pas de ce biais. Si $C_{i,j}$ représente le terme ligne i et colonne j de la matrice de confusion, alors ce score est défini par: $S = \sum_i \frac{C_{i,i}}{\sum_j C_{i,j}}$. Un score de 0 indique un échec total, un score de 1 ou 100% indique un reconnaissance de tous les objets. Le hasard obtient une performance moyenne de $1/C$, où C est le nombre de classes.

4.7.3 Production du vocabulaire visuel

Le vocabulaire est calculé avec l'algorithme présenté section 4.6.3. Le seuil de la corrélation normée centrée est fixé à 0.8 (les valeurs possibles variant de -1 à 1, 1 étant le score de corrélation maximale), ce qui correspond à un angle de 40 degrés entre les deux descripteurs de régions locales. La pyramide d'échelles a 9 niveaux, et le rapport entre deux échelles successives est de 0.91. L'algorithme de mean-shift produit itérativement 4000 mots de vocabulaire visuel. Le rayon des boules de mean-shift est fixé à 0.8. La figure 4.10 présente des mots visuels très fréquents et très rares produits à partir des images infra-rouge.

4.7.4 Sélection de primitives

Dans cette section, nous comparons les différentes méthodes de sélection de primitives pour choisir laquelle utiliser dans la hiérarchie de classifieurs binaires. Nous utilisons des données non normalisées et la performance d'un classifieur SVM un-contre-un pour comparer les différentes méthodes. La figure 4.11 présente la performance de classification en fonction du pourcentage de primitives considérées.

Nous pouvons tout d'abord observer que le vocabulaire visuel contient des primitives inutiles ou redondantes, car la sélection de primitives par taux d'information mutuelle atteint

Figure 4.10: Au dessus (resp. en dessous) les 80 mots visuels les plus (resp. moins) fréquents dans la base de données infra-rouge.

la performance maximale avec seulement 40% des primitives sur la base de données infra-rouge, et 10% des primitives sur la base de données visible.

Comparons maintenant les différentes courbes. La sélection de primitives par wrapper est la plus efficace, car l'objectif optimisé est justement la performance de classification. Contrairement aux autres méthodes de sélection de primitives, la courbe croît très rapidement. Par contre, elle a une complexité quadratique quand les autres ont une complexité linéaire. Les sélections de primitives par rapport des côtes ou par fréquence donnent des performances moins bonnes que la sélection de primitives aléatoire, qui est notre référence. Nous ne considérerons donc pas ces deux méthodes de sélection de primitives. Dans la base de données visible, la sélection de primitives par les coefficients du SVM donnent aussi une performance inférieure à une sélection aléatoire, la sélection de primitives par SVM n'est donc pas intéressante avec nos données. Ce n'est pas le cas de la sélection de primitives par taux d'information mutuelle, dans tous les cas elle se comporte mieux qu'une sélection de primitives aléatoire. En effet, sur la base visible le gain est de 20% avec 1% des primitives, et de 25% avec 4% des primitives, sur la base infra-rouge le gain est de 15% avec 4% des primitives.

On peut conclure que le vocabulaire est très redondant, et qu'il est possible de réduire sa taille sans perte significative de performance. Parmi les méthodes de sélection de primitives, la méthode par wrapper donne les meilleures performances; et parmi les méthodes linéaires en temps de calcul, la sélection de primitives par taux d'information mutuelle est la plus efficace. Du fait de la complexité quadratique de la méthode par wrapper, et du temps de calcul prohibitif qui en découle, nous ne considérerons par la suite que la sélection de primitives par taux d'information mutuelle.

4.7.5 Sélection de primitives pour hiérarchie de classifieurs binaires

Dans cette section, nous mesurons l'intérêt de la sélection de primitives pour les hiérarchies de classifieurs binaires. Notre but est de comparer ces hiérarchies de classifieurs binaires aux

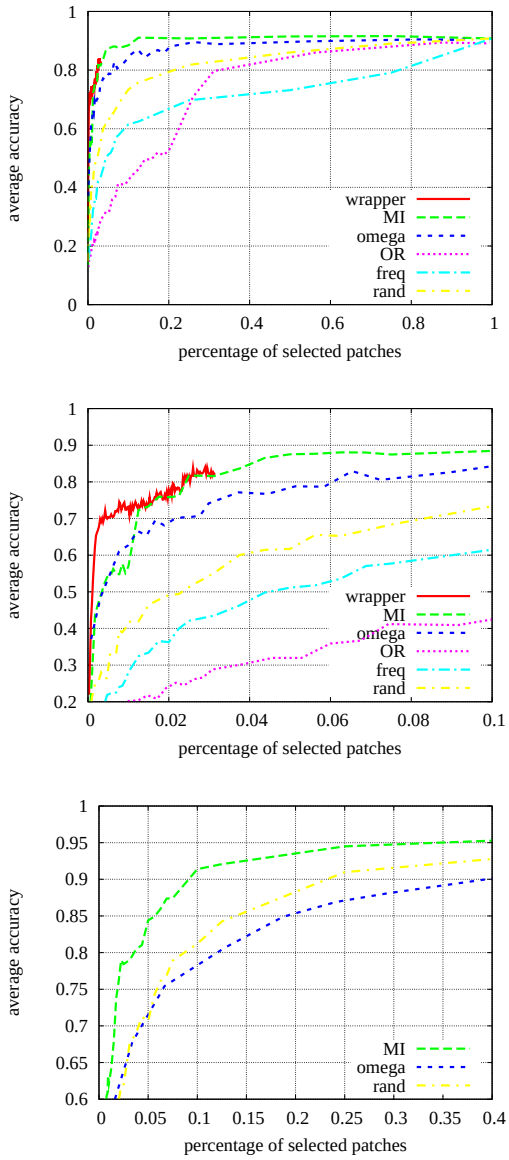


Figure 4.11: Performance des différentes méthodes de sélection de primitives. Lignes 1 et 2: base de données visible, aperçu global et zoom sur une région. Ligne 3: base de données infra-rouge.

classifieurs plus classiques. Nous utilisons deux types de classifieurs hiérarchiques: le premier type est calculé par l'algorithme de Rajan [74] (voir section 4.4), le second regroupe les classes dans les différents noeuds aléatoirement, ce qui permet de mesurer l'intérêt d'utiliser l'algorithme proposé par Rajan. La figure 4.12 présente la performance de classification avec les deux types d'arbres, les classifieurs SVM un-contre-un, SVM un-contre-tous et le classifieur bayésien naïf. L'arbre appris automatiquement est présenté figure 4.16.

Nous observons tout d'abord que les classifieurs un-contre-un et un-contre-tous sont plus performants que la hiérarchie de classifieurs binaires quand l'intégralité des primitives est utilisée. En effet, la hiérarchie de classifieurs binaires prend peu de décisions, et la moindre erreur est responsable d'une mauvaise classification. Au contraire, le classifieur un-contre-un possède une grande redondance qui permet de ne pas être sensible à la moindre erreur. Le classifieur un-contre-tous est lui aussi robuste à de petites erreurs: tant que la véritable classe obtient le score maximal, peut importe si les autres classes remportent une compétition.

Cependant, la zone de fonctionnement qui nous intéresse n'est pas celle qui utilise toutes les primitives, mais un nombre très réduit de primitives, afin de réduire considérablement le temps de calcul. Dans ce cas, la hiérarchie de classifieurs binaires est plus performante que les autres classifieurs. Par exemple, sur la base de données infra-rouge, la hiérarchie de classifieurs binaires est plus performante que le classifieur un-contre-un de 10% avec 1% des primitives — soit 40 mots visuels, 4% avec 3% des primitives, et la performance est identique avec 5% des primitives. Dans la base visible, le gain est de 18% avec 1% des primitives, 8% avec 3%, et la performance est identique avec 4% des primitives.

On remarque aussi que l'utilisation d'arbres adaptés aux classes influence nettement la classification, car l'arbre aléatoire produit des résultats de classification significativement plus mauvais que l'arbre appris avec l'algorithme de Rajan [74] (figure 4.12, dernière ligne).

De ces observations on peut conclure que la hiérarchie de classifieurs binaires est plus performante que les classifieurs SVM standard un-contre-un et un-contre-tous avec un nombre restreint de mots visuels.

4.7.6 Normalisation des données

Dans cette section nous étudions l'effet de la normalisation des données utilisées par les SVM linéaires sur la performance de classification. Nous comparons les représentations suivantes: données brutes, transformations linéaires par primitive, transformations linéaires par image, binarisation simple, binarisation adaptative. Les différents résultats sont résumés dans le tableau 4.1.

La transformation linéaire par image et par primitive sont deux transformations populaires [62]. Nous observons que par rapport aux données brutes non normalisées, la normalisation par primitive et la binarisation simple augmentent la performance et la normalisation par image la diminue. La binarisation adaptative que nous proposons surpasse toutes les normalisations ci-dessus.

Sur la base infra-rouge, la binarisation adaptative et la binarisation simple donnent la même amélioration: +7% par rapport aux données brutes. Sur la base en lumière visible, la binarisation adaptative est +9% meilleure que les données brutes, et +4% meilleure que la binarisation simple. La figure 4.13 compare les deux types de binarisation pour les classifieurs un-contre-un ou pour les hiérarchies de classifieurs binaires sur la base visible: pour les deux classifieurs, la binarisation adaptative permet d'améliorer significativement la performance.

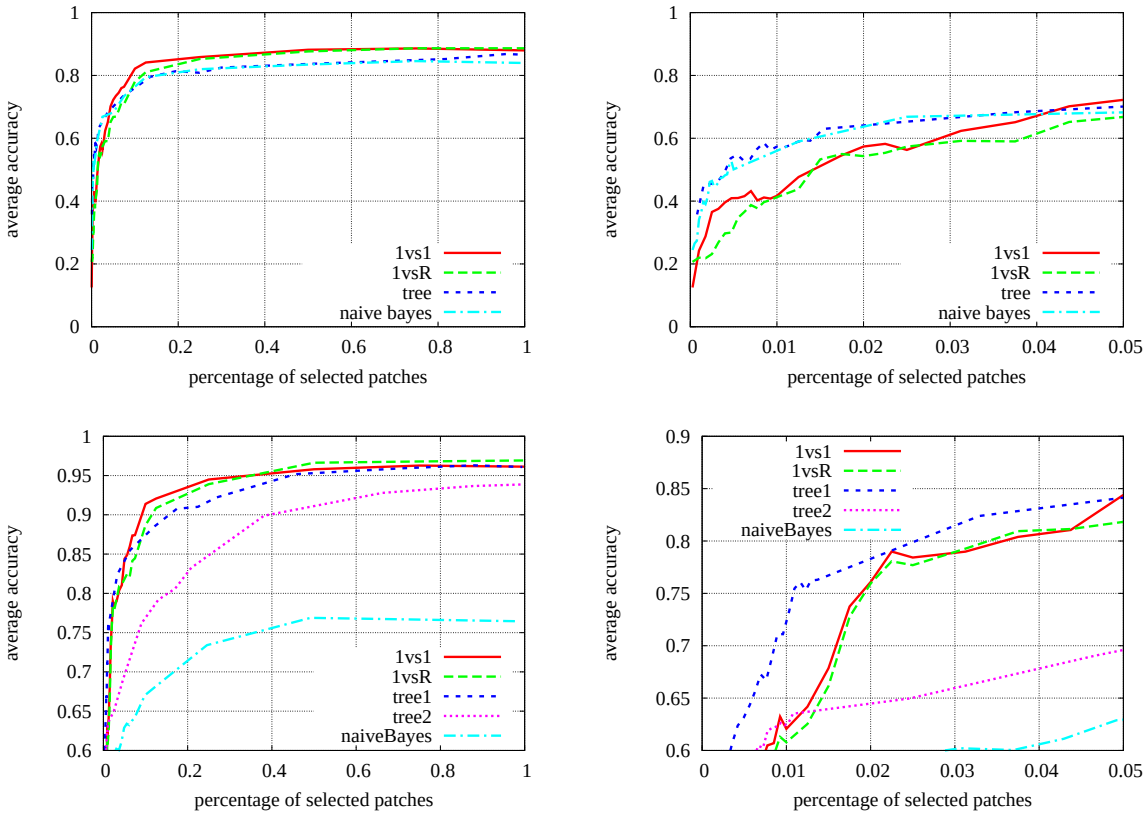


Figure 4.12: Influence de la méthode de classification multi-classes. Haut: base de données visible (graphe complet et zoom). Bas: base de données infra-rouge (graphe complet et zoom). Les graphes indiquent la performance de classification multi-classes en fonction du pourcentage de mots visuels utilisés.

Transformation	*Performance*
Données brutes	**83%**
Trans. lin. par primitive	87%
Trans. lin. par image	78%
Binarisation simple	88%
Binarisation adaptative	92%
Transformation	*Performance*
Données brutes	**89%**
Trans. lin. par primitive	93%
Binarisation simple	96%
Binarisation adaptative	96%

Table 4.1: Influence des méthodes de normalisation des données. Haut: base de données visible. Bas: base de données infra-rouge.

Rang	1	2	3	4	5	6	7
Base visible	300	500	2500	2250	6000	200	150
Base infra-rouge	0	0	0	0	0	6	0

Table 4.2: Seuils de binarisation adaptative automatiquement choisis pour les sept primitives les plus importantes (critère: taux d'information mutuelle). Les seuils choisis par binarisation adaptative sont très élevés pour la base visible et très faibles pour la base infrarouge, d'où la similarité des performances entre la binarisation adaptative et la binarisation standard (seuil 0) sur la base infrarouge.

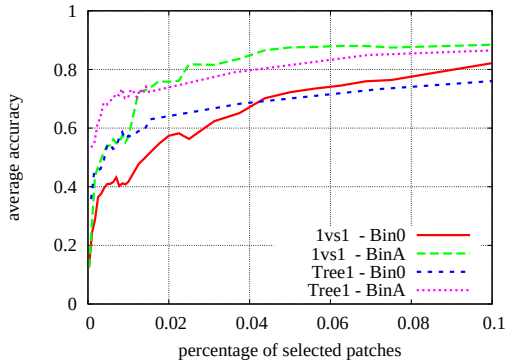


Figure 4.13: La binarisation adaptative améliore la performance du classifieur un-contre-un et de la hiérarchie de classifieurs binaires.

Il est intéressant d'observer pourquoi la binarisation simple et la binarisation adaptative donnent la même performance sur la base infra-rouge, alors que la binarisation adaptative est largement meilleure que la binarisation standard (seuil 0) sur la base visible. Le tableau 4.2 indique quels seuils de binarisation ont été automatiquement sélectionnés par la binarisation adaptative sur les deux bases. Dans la base visible, les seuils de binarisation sont très élevés, donc la binarisation simple est moins informative, ce qui se traduit dans les performances observées. A l'inverse, dans la base infra-rouge, les seuils choisis automatiquement sont bas, donc la binarisation simple est une bonne approximation de la binarisation optimale, ce qui une fois de plus se traduit dans les résultats observés.

La binarisation adaptative est donc un choix préférable à la binarisation simple, car elle est une généralisation de celle-ci, et qu'elle permet de choisir les seuils de binarisation adaptés à une base de données, et ce pour chacune des primitives.

4.7.7 Base infra-rouge: performance en situation de jamais-vu

Situation de jamais-vu

La performance mesurée sur dans la section précédente sur la base infra-rouge se faisait par cross-validation. Dans cette section, nous utilisons deux jeux de données bien séparés.

L'apprentissage se fait à partir des images d'une seule séquence vidéo par véhicule. Les véhicules sont sur une route, et font demi-tour sur eux-mêmes afin de capturer les variations dues au changement d'orientation. Des images typiques de cette séquence peuvent être observées figure 4.5. L'évaluation se fait dans des conditions différentes, cela permet d'évaluer le système dans une situation de terrain, où la reconnaissance ne se fait pas nécessairement dans les conditions d'apprentissage. Par exemple, les véhicules sont observés sur la même route mais dans des conditions thermiques très différentes (enregistrements de nuit), dans un pré, avec des occultations, etc. Ces conditions d'observation différentes peuvent être observées figures 4.4, bas et 4.6.

La base d'apprentissage contient 140 images de véhicules et 750 images de fond, la base de test contient 307 images de véhicules et 750 images de fond. L'apprentissage se fait à 100 mètres, où une AX est vue à 70 pixels de hauteur. Lors du test, les véhicules vont d'une distance de 100 à 600 mètres, ce qui veut dire que la hauteur de l'AX varie entre 70 et 25 pixels de hauteur.

Vocabulaire visuel

Nous apprenons le vocabulaire visuel avec la méthode proposée par Jurie [45]. Nous apprenons un total de 4000 mots visuels décrits avec leurs niveaux de gris, ces mots sont représentés figures 4.14 et 4.15. Les mots sont classés des plus fréquents aux moins fréquents dans les images. En raison du manque de robustesse aux translations/rotations/changements d'échelle du descripteur niveaux de gris, on constate que de nombreux motifs ont une apparence quasi-identique, à une petite transformation géométrique près.

Sans surprise, les mots visuels les plus fréquents correspondent aux signaux de plus basse fréquence, le mot le plus fréquent étant justement la région uniforme. Cela ne se produit pas avec les détecteurs de points d'intérêt classiques car ils ne détectent que les régions présentant une certaine variation de signal.

Hiérarchie de classifieurs binaires

L'arbre appris automatiquement sur les données d'apprentissage avec l'algorithme de Rajan [74] (voir section 4.4) produit l'arbre présenté figure 4.16. On constate que les classes les plus différentes des autres en sont séparées dès le début: le premier noeud sépare le fond des véhicules, et le second sépare la fourgonnette des véhicules légers.

Performance

La matrice de confusion obtenue avec la méthode proposée est présentée figure 4.17. Elle est normalisée de manière à être exprimée en pourcentages. On constate tout d'abord qu'il y a une bonne séparation entre le fond et les véhicules: malgré la quantité importante d'images de fond, environ 99% des images de fond sont reconnues comme telles. Ces images étant très nombreuses, nous ne représentons qu'une image sur 12 dans la figure 4.18. Les 4 images

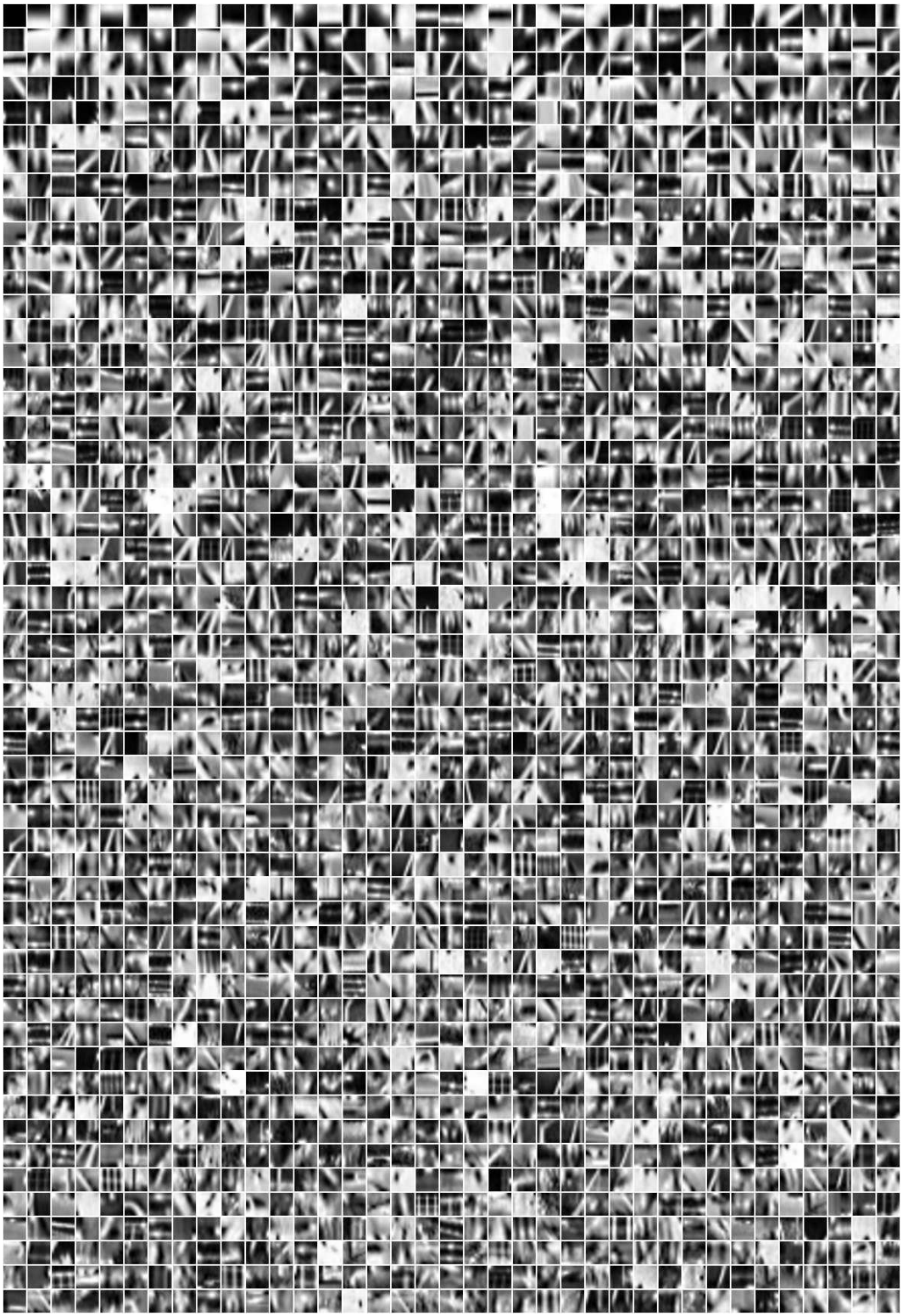


Figure 4.14: Vocabulaire appris sur la base infra-rouge Bertin: les 1998 premiers éléments, classés des plus fréquents au moins fréquents dans la base d'apprentissage.

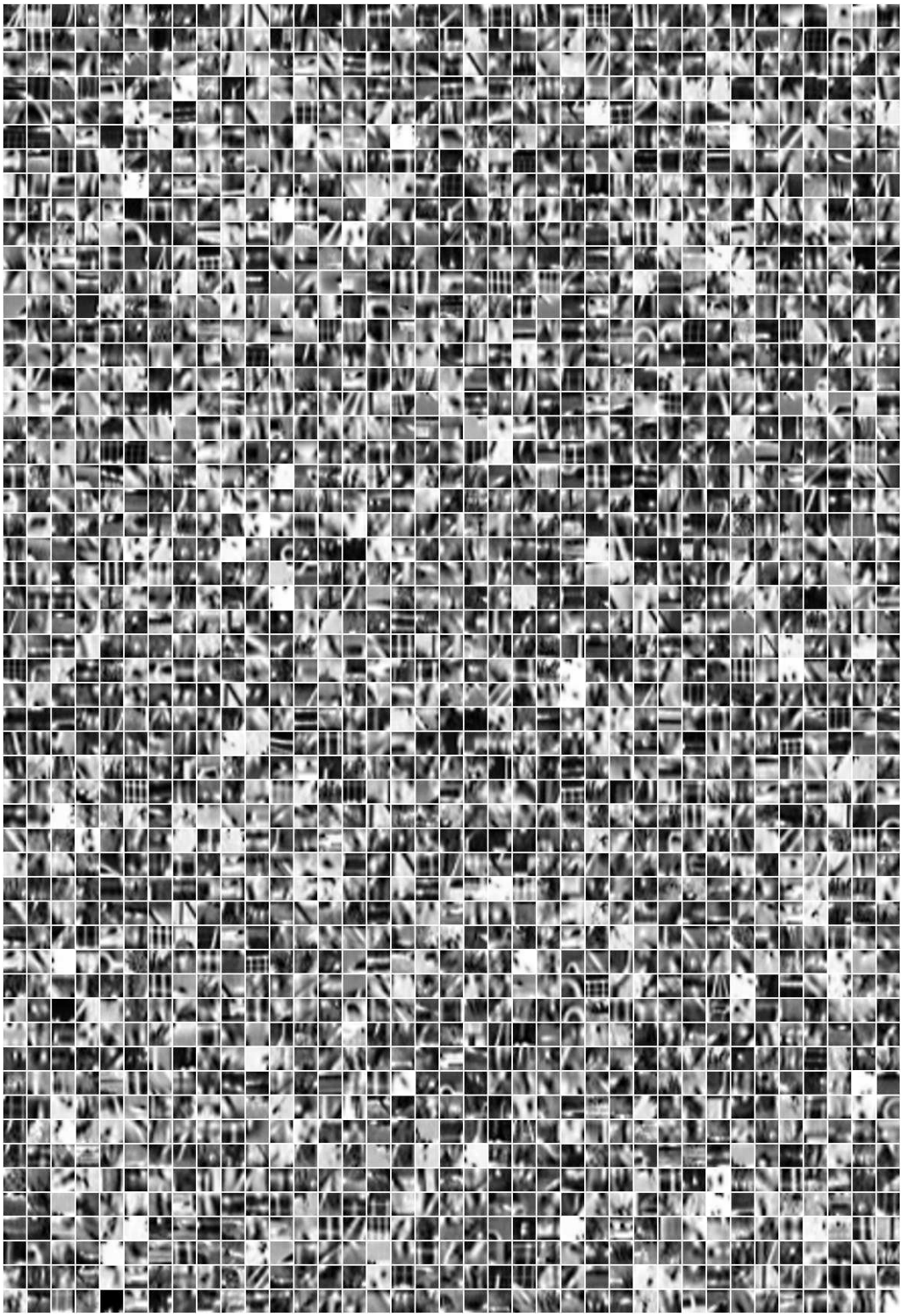


Figure 4.15: Vocabulaire appris sur la base infra-rouge Bertin: les 1998 éléments suivants, classés des plus fréquents au moins fréquents dans la base d'apprentissage.

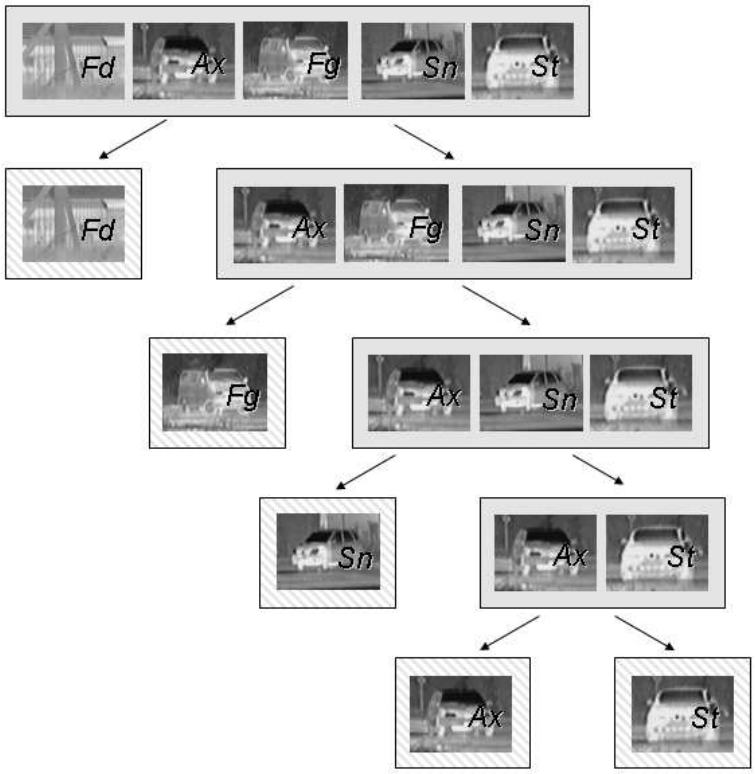


Figure 4.16: Hiérarchie de classes apprise sur la base Bertin. Les feuilles sont indiquées par des hachures. Fd: fond, Ax: Citron AX, Fg: Fourgonnette, Sn: Renault Scenic, St: Seat Ibiza. Les classes séparées du reste en premier sont celles qui s'en différencient le plus: le fond puis la fourgonnette.

mal classées sur 1500 (soit 0.3%) se répartissent entre les catégories "AX" (figure 4.19), "Fourgon" (figure 4.20) et "Scénic" (figure 4.21). De plus, une faible quantité d'images de véhicules sont pris pour des fonds, en moyenne $(13+7+7+8)/4 = 8.8\%$. Ces images sont représentées figures 4.22, 4.26, 4.30 et 4.34. Dans la très grande majorité des cas, ces images contiennent des véhicules très occultés par du fond, ce qui explique la confusion.

Analysons maintenant les erreurs de classification entre véhicules. Une grande majorité des "fourgons" (figure 4.28) et des "Scénic" (figure 4.32) sont reconnus comme tels. Par contre, seulement environ la moitié des "AX" (figure 4.23) et des "Seat" (figure 4.38) sont reconnues comme telles, en raison justement d'une confusion entre ces deux catégories (figures 4.25 et 4.35). Rappelons que la performance du hasard est de 20% (une chance sur 5). Les autres confusions sont assez faibles, voir nulles, comme on peut le voir sur les figures 4.24, 4.27, 4.29, 4.31, 4.33, 4.36 et 4.37.

Ces performances sont remarquables quand on considère la faible quantité d'images

d'apprentissage (35 images en moyenne par véhicule), et la grande différence entre les conditions d'apprentissage et de test (et les différences de type d'image qui en résultent)

La performance globale du système en situation de jamais vu est de 71% (moyenne de la diagonale de la matrice de confusion normalisée). La performance de la méthode de base que nous avions implémenté à l'origine est de 40%, et le hasard est de 20%. La méthode d'origine utilise une représentation globale des images: une grille multi-échelle est plaquée sur les imagettes à analyser, et dans chaque cellule de la grille on calcule des convolutions entre une banque de filtres et la cellule. Les filtres utilisés sont les dérivées de gaussiennes du premier ordre (2 dérivées) et second ordre (3 dérivées) soit 5 filtres. La grille multi-échelle considérée est faite de 1,2 et 3 subdivisions, et donc les trois niveaux contiennent 1,4 et 9 cellules, soit 14 au total. Le vecteur descripteur contient les statistiques (moyenne et variance) de chaque filtre dans chaque cellule, et a donc une dimension de $2*5*14 = 140$. Chaque image est décrite par un tel vecteur, et un SVM est appris pour séparer les différentes classes.

Classe "fond". Les prédictions pour les images de la classe "fond" sont présentées figures 4.18, 4.19, 4.20 et 4.21.

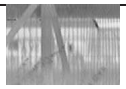
	 Fond	 AX	 Fourgon	 Scenic	 Seat
	99	0	0	0	0
	13	45	0	13	27
	16	1	79	0	2
	7	13	0	75	3
	8	25	7	2	57

Figure 4.17: Matrice de confusion sur une tâche de reconnaissance en situation de jamais-vu sur la base Bertin. Le nombre situé à l'intersection de la ligne *i* et de la colonne *j* indique le pourcentage des exemples de la catégorie *i* qui ont été prédits comme membres de la catégorie *j*

Figure 4.18: Exemples de la classe "fond" reconnus comme tels. On ne montre ici qu'un douzième des images correctement reconnues.

Classe "AX". Les prédictions pour les images de la classe "AX" sont présentées figures 4.22, 4.23, 4.24, 4.25.

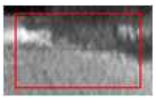


Figure 4.19: Intégralité des exemples de la classe "fond" pris pour des "AX".

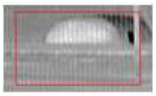


Figure 4.20: Intégralité des exemples de la classe "fond" pris pour des "fourgons".

Figure 4.21: Intégralité des exemples de la classe "fond" pris pour des "Scénic".

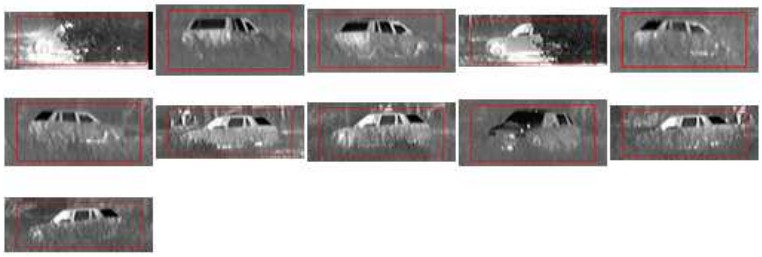


Figure 4.22: Intégralité des exemples de la classe "AX" pris pour des "fonds".

Classe "Fourgon". Les prédictions pour les images de la classe "Fourgon" sont présentées figures 4.26, 4.27, 4.28, 4.29.

Figure 4.23: Intégralité des exemples de la classe "AX" reconnus comme tels.

Figure 4.24: Intégralité des exemples de la classe "AX" pris pour des "Scénic".

Classe "Scenic". Les prédictions pour les images de la classe "Scenic" sont présentées figures 4.30, 4.31, 4.32, 4.33.

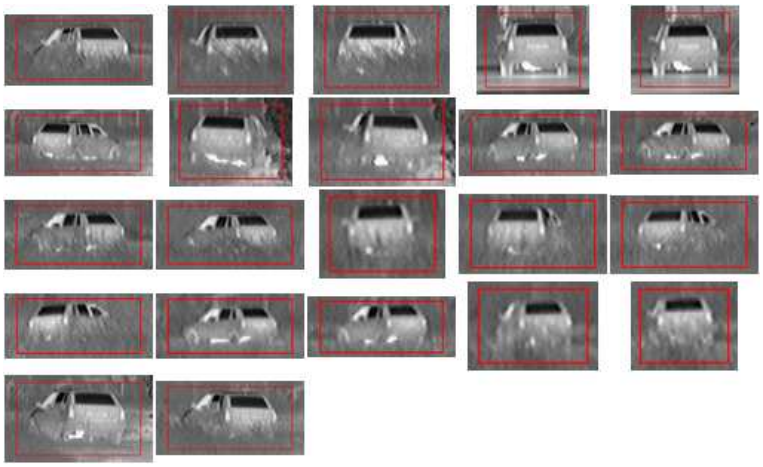


Figure 4.25: Intégralité des exemples de la classe "AX" pris pour des "Seat".

Figure 4.26: Intégralité des exemples de la classe "Fourgon" pris pour des "fonds".

Figure 4.27: Intégralité des exemples de la classe "Fourgon" pris pour des "AX".

Classe "Seat". Les prédictions pour les images de la classe "Seat" sont présentées figures 4.34, 4.35, 4.36, 4.37, 4.38.

Figure 4.28: Intégralité des exemples de la classe "Fourgon" reconnus comme tels.

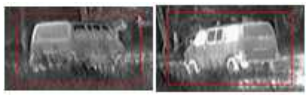


Figure 4.29: Intégralité des exemples de la classe "Fourgon" pris pour des "Seat".

Figure 4.30: Intégralité des exemples de la classe "Scénic" pris pour des "fonds".

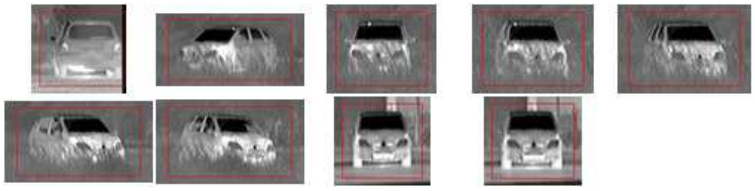


Figure 4.31: Intégralité des exemples de la classe "Scénic" pris pour des "AX".

Figure 4.32: Intégralité des exemples de la classe "Scénic" reconnus comme tels.

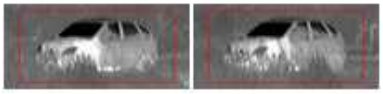


Figure 4.33: Intégralité des exemples de la classe "Scénic" pris pour des "Seat".

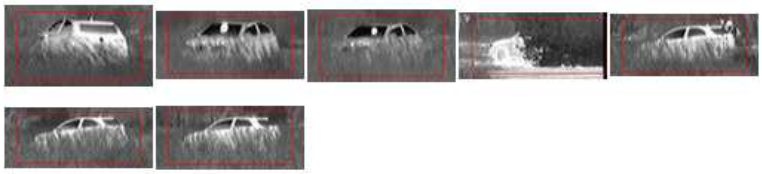


Figure 4.34: Intégralité des exemples de la classe "Seat" pris pour des "fonds".

Figure 4.35: Intégralité des exemples de la classe "Seat" pris pour des "AX".

Figure 4.36: Intégralité des exemples de la classe "Seat" pris pour des "fourgons".

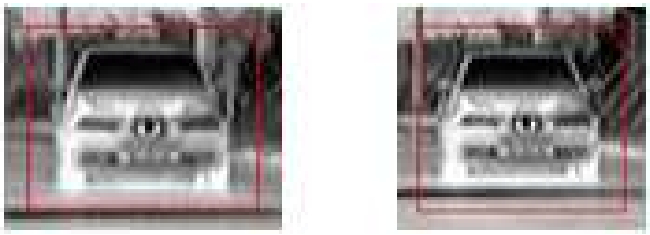


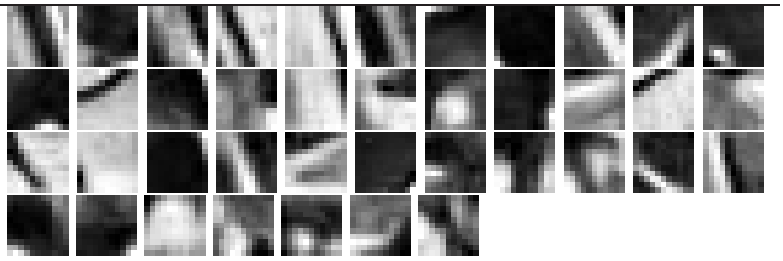
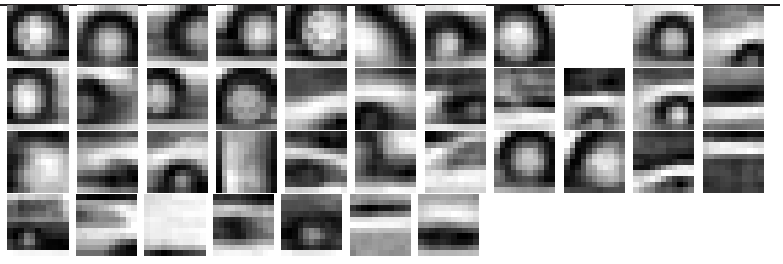
Figure 4.37: Intégralité des exemples de la classe "Seat" pris pour des "Scénic".

Figure 4.38: Intégralité des exemples de la classe "Seat" reconnus comme tels.

4.7.8 Observation de la sélection de primitives

Dans cette section nous observons quelles primitives sont sélectionnées aux différentes noeuds d'un arbre. La base considérée est la base Xerox7. L'arbre appris automatiquement par l'algorithme de Rajan [74] est présenté figure 4.39. Les 40 meilleures primitives sélectionnées par noeud sont présentées dans le tableau 4.3.

On constate que les primitives qui servent à séparer de grands ensembles de classe sont de basse fréquence, et qu'elles traduisent en particulier la séparation entre deux régions, ou une orientation de gradient (voir ligne 1 par exemple). En effet, pour pouvoir différencier deux larges groupes de classes, les primitives utilisées doivent être génériques. Pour différencier des groupes dont l'un ne contient qu'une seule classe, les primitives sont de plus haute fréquence. Car ces primitives spécifiques au groupe d'une seule classe permettent de différencier cette classe de celles de l'autre groupe. Par exemple ligne 2, de nombreux mots visuels correspondent à des parties de voitures, ce qui permet de différencier la classe voiture des classes Visage et Fond. C'est aussi le cas pour la ligne 7, les mots visuels représentent des livres (vus par la tranche) et permettent de différencier les livres des téléphones. De même, ligne 3, de nombreux mots visuels représentent des parties de visage (menton, yeux et arcade, visage complet) qui permettent de séparer les visages et le fond. Citons encore la ligne 5, où des mots visuels représentant des parties de roues de vélos permettent de différencier les vélos des chevaux et motos. Enfin, la ligne 6 contient des mots visuels représentant des roues de motos, ce qui permet de les différencier des chevaux. On observe dans la ligne 6 des mots visuels provenant de visages, ce qui peut paraître très surprenant au premier abord pour séparer les chevaux des motos. L'observation des vecteurs descripteurs montre que ces mots sont souvent détectés dans la classe Cheval en raison du seuil de corrélation utilisé (0.8).

Ligne	Classes	Meilleurs mots visuels
1	{Visage Voiture Fond} **vs** {Cheval Moto Vélo Livres Téléphones}	
2	{Voiture} **vs** {Visage Fond}	

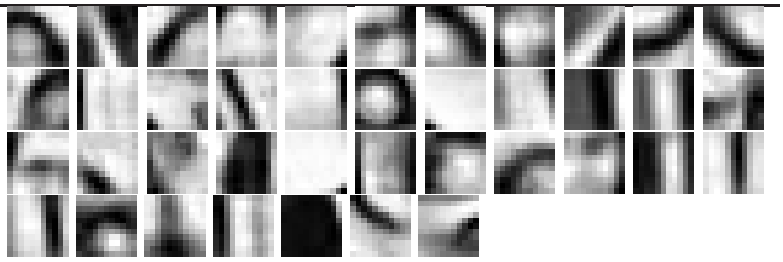
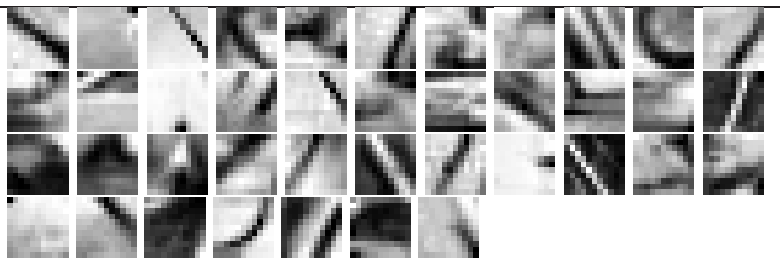
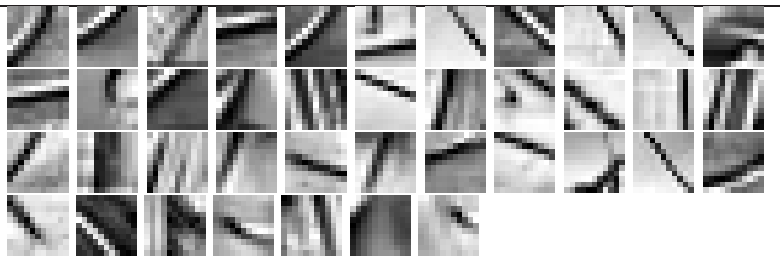
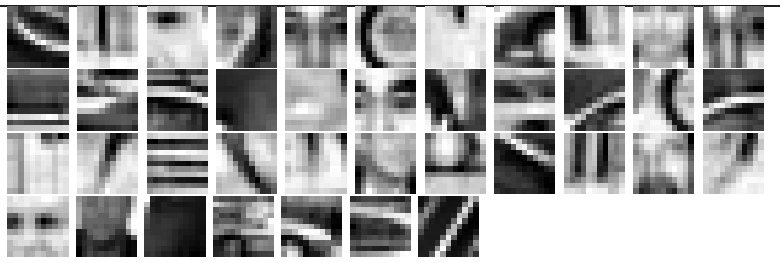
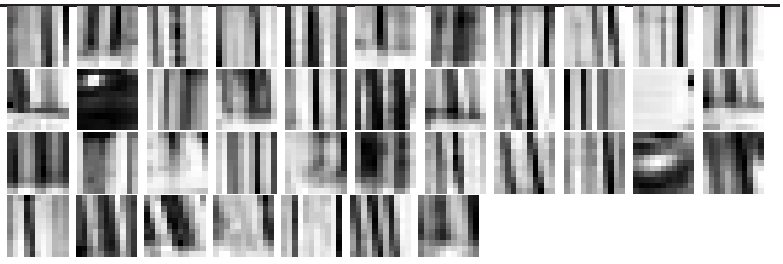
3	{Visage} **vs** {Fond}	
4	{Cheval Moto Vélo} **vs** {Livres Téléphones}	
5	{Cheval Moto} **vs** {Vélo}	
6	{Cheval} **vs** {Moto}	
7	{Livres} **vs** {Téléphones}	

Table 4.3: Les 40 meilleurs mots visuels sélectionnés par taux d'information mutuelle pour séparer les groupes de classe de la première colonne.

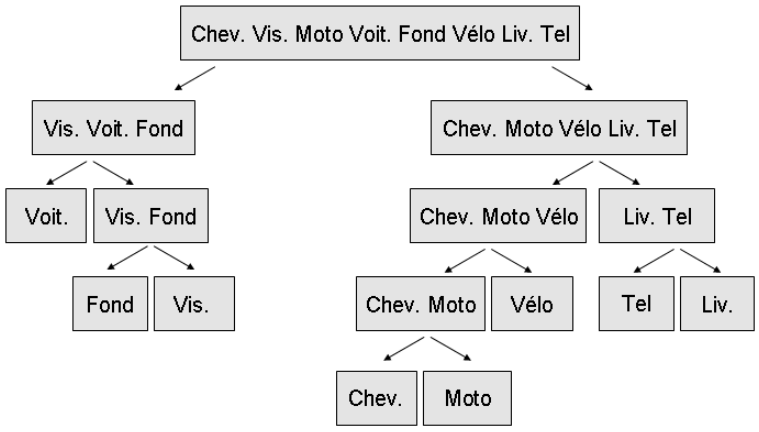


Figure 4.39: Hiérarchie de classes apprise sur la base Xerox. Chev: Cheval, Vis: Visage, Moto: Moto, Voit: Voiture, Fond: fond (divers), Vélo: Vélo, Liv: Livre, Tel: Téléphone.

4.8 Conclusion

Nous nous sommes intéressés au problème de reconnaissance de modèles de véhicules par des approches sacs-de-mots, et au compromis entre performance de classification et nombre de mots visuels utilisés par le modèle — qui influencent directement le temps de calcul.

Nous avons pour cela proposé une nouvelle méthode de normalisation des données, basée sur la maximisation de l'information mutuelle entre les occurrences des mots visuels et les catégories d'objets. Nous avons aussi proposé un algorithme de sélection de primitives (les mots visuels) adaptée aux hiérarchies de classifieurs binaires, qui surpasse significativement le classifieur SVM multi-classes standard un-contre-un quand la quantité de primitives à utiliser est faible.

Nous avons montré que nous obtenons de bonnes performances de classification sur une base de données infra-rouge qui contient différents modèles de véhicules et du fond, ainsi que sur une base de données visible de catégories d'objets génériques.

FIFTH CHAPTER

Comparaison d'objets jamais vus

5.1 Résumé du chapitre

Dans ce chapitre, nous proposons et évaluons un algorithme qui apprend une mesure de similarité visuelle entre images, destinée à comparer des objets jamais vus lors de la phase d'apprentissage. Cette mesure est apprise à partir de paires d'images labélisées "identiques" ou "différentes". Ces labels sont bien moins informatifs que les labels de classes usuels (par exemple "voiture de type R5") mais plus facile à obtenir. L'algorithme proposé apprend les différences caractéristiques entre des descripteurs de régions locales échantillonnées dans des paires d'images "identiques" ou "différentes". Ces différences sont quantifiées par un ensemble d'arbres extrêmement aléatoires, et la mesure de similarité entre les deux images est calculée à partir de ces différences quantifiées. Les arbres extrêmement aléatoires sont rapides à apprendre, robustes en raison de l'information redondante qu'ils contiennent, et il a été montré qu'ils sont de très bons quantificateurs. De plus, les arbres sont capables de combiner efficacement différentes caractéristiques mesurées dans les images (descripteurs SIFT et informations géométriques). Nous évaluons notre mesure de similarité innovante sur quatre bases très différentes, et surpassons toujours significativement les résultats précédemment publiés.

5.2 Introduction

Dans les chapitres précédents, nous nous sommes intéressés à la reconnaissance visuelle de catégories d'objets. Nous avons considéré une représentation par sac-de-mots des images, et avons montré que celle-ci obtient aujourd'hui les meilleures peformances de l'état de l'art. Dans ce chapitre, nous nous intéressons à la reconnaissance visuelle *d'instances* d'objets. Nous expliquerons plus loin qu'une simple représentation d'image par sac-de-mots ne permet pas de reconnaître des instances d'objets, car le principe de cette représentation est la quantification (et donc *l'approximation*) de l'apparence des parties locales, hors ce sont des *détails précis* qui permettent d'identifier les objets. Il faut donc considérer une autre représentation d'image spécifique à la reconnaissance d'instance d'objet, c'est l'objet du présent chapitre. De plus, nous nous plaçons dans le cas le plus difficile où l'objet que l'on cherche à reconnaître n'a été vu qu'une seule fois auparavant.

On constate que les humains reconnaissent facilement des objets qu'ils n'ont vu qu'une seule fois. Il est possible de reconnaître une personne vue une seule fois, malgré l'apparition de différences: vêtements, coiffure, lunettes, expression, ... De même, il est possible de reconnaître une voiture vue une seule fois, malgré l'apparition de différences: point de vue, lumière, couleur, ... (voir figure 5.1). C'est parce que nous avons une certaine connaissance de notre environnement, et des connaissances sur les personnes et les voitures en particulier, aussi une seule vue d'un nouvel objet d'une catégorie générique connue nous permet de le reconnaître plus tard.

Comparer deux images – et plus généralement comparer deux exemples – dépend beaucoup de la définition d'une bonne mesure de similarité. Les fonctions standard (par exemple la distance euclidienne dans l'espace de représentation des images) sont souvent trop génériques et ne permettent pas de prendre en compte les *connaissances spécifiques au domaine*. Aussi proposons nous d'apprendre une mesure de similarité qui prend en compte les informations spécifiques au domaine considéré. Cette mesure ne peut être dérivée simplement d'une description d'image par sac-de-mots, car comme nous l'indiquons dans la section 5.3.1, les représentations par sac-de-mots sont adaptées aux catégories génériques d'objets, mais pas à l'identification précise d'instances d'objets.

De plus, nous proposons d'apprendre cette mesure à partir de *contraintes d'équivalence*. Les contraintes d'équivalence sont des paires d'exemples d'apprentissage représentant des objets identiques ou différents. Une paire d'images n'est donc plus labélisée "modèle de voiture X et modèle de voiture Y" mais seulement "identique" ou "différent". Cette représentation est plus difficile à utiliser, car elle est moins informative que des labels complets: des paires peuvent être produites à partir des labels complets, mais le contraire n'est pas possible. Dans de nombreuses applications, il est plus facile d'obtenir des contraintes d'équivalence que des labels complets. Par exemple, si l'espace des catégories est très grand, il est plus facile de dire si deux objets appartiennent à la même catégorie ou non, plutôt que de déterminer précisément la catégorie de deux objets. De plus, lorsque des objets de catégorie inconnue sont présentés au système (modèles de voitures X et Y jamais vus) il est impossi-

ble de prédire leur catégorie (puisqu'elle n'a jamais été apprise) mais il est au moins possible de dire si les les catégories sont identiques ou différentes.

Dans notre étude, nous utilisons cette mesure de similarité pour *l'identification visuelle d'objets jamais appris*. Étant donné un ensemble d'apprentissage de paires d'images labélisées "identiques" ou "différentes", nous devons décider si deux objets jamais vus auparavant sont identiques ou différents (voir figure 5.2).

5.2.1 Travaux similaires

L'apprentissage d'une mesure de similarité pour comparer des données est un domaine très actif, sur lequel de nombreuses équipes se sont focalisées durant ces dernières années. La plupart des travaux consiste à transformer l'espace de représentation des images afin d'obtenir un nouvel espace de représentation dans lequel une simple distance permet de comparer les exemples.

Cette transformation est généralement basée sur la distance de Mahanalobis, qui s'exprime généralement par $d(x,y) = (x-y)^t A(x-y)$, voir [55, 96, 38, 82, 4, 36, 92]. Des approches variées permettent d'estimer la matrice A, différences s'expliquant par diverses formulations de la fonction objectif à minimiser. Dans [96, 38] la fonction objectif tend à regrouper les exemples de même classe, et à éloigner les exemples de classes différentes. [38] maximise une variante stochastique du score leave-one-out d'un classifieur plus proche voisin. Dans [92], la fonction objectif sépare les exemples différents en maximisant la marge de séparation, avec un classifieur plus proche voisin. Dans [82] aussi la marge est maximisée. Dans [4] la matrice A est calculée avec des *chunklets*, qui sont des ensembles de contraintes d'équivalence issues de la base d'apprentissage. La transformation d'un espace à l'autre peut aussi être apprise sans utiliser de fonction explicitement définie, comme dans [13] où un réseau de neurones est utilisé pour sa robustesse aux déformations géométriques. Si les données considérées sont des images, des fonctions plus spécifiques encodant des transformations habituelles d'objets (point de vue, lumière, ...) peuvent être intégrées [61, 30, 22].

Malheureusement, aucune de ces méthodes n'est parfaitement adaptée à l'identification visuelle dans des images. Contrairement aux problèmes de reconnaissance de formes classiques, les informations contenues dans des images sont sujettes à des transformations complexes, telles que des occultations, modifications de point de vue, d'échelle, qui ne peuvent être modélisées facilement par des transformations des données linéaires, quadratiques, ou polynomiales en général.

La méthode habituelle pour résoudre ce problème de transformations complexes est de représenter les images par un ensemble de parties locales (fenêtres en niveaux de gris, descripteurs SIFT [56] ou autres), car *localement* certaines parties locales subissent peu de modifications. Cette stratégie a été utilisée par [24, 25]; leur algorithme consiste à apprendre quelles parties locales permettent de décider si deux objets sont similaires ou différents. La méthode de *chopping* proposée par [31] est aussi intéressante: elle consiste à créer une grande quantité de séparations binaires dans l'espace des images, puis à combiner les classifieurs entraînés sur ces séparations. Cependant leur approche nécessite d'avoir des

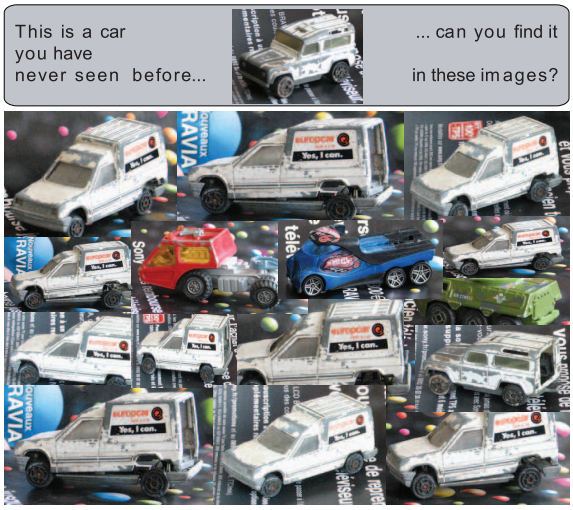


Figure 5.1: Notre connaissance des voitures en général nous permet de reconnaître un nouveau modèle (jamais vu auparavant), malgré des modifications de point de vue, lumière, fond. Cette étude propose un algorithme qui effectue une telle identification d'objets jamais vus auparavant.

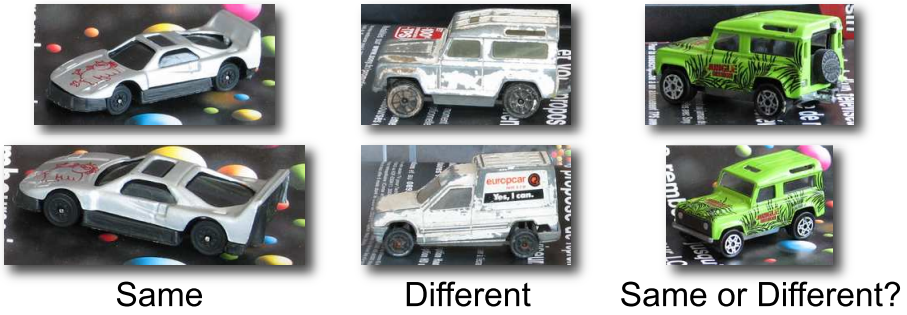


Figure 5.2: Étant données des paires d'images labélisées "identiques" ou "différentes", pouvons nous apprendre une mesure de similarité indiquant si deux images représentent des objets identiques ou différents? Cette mesure de similarité doit être robuste aux modifications de point de vue, de fond, d'illumination, et surtout elle doit pouvoir comparer deux objets *jamais vus auparavant*.

données totalement labélisées lors de l'apprentissage, et non pas des paires "identiques" et "différentes".

Inspirés par ces travaux, et en particulier par les travaux de Ferencz et al [25], nous proposons un nouvel algorithme pour l'apprentissage d'une mesure de similarité entre deux images d'objets jamais vus. L'apprentissage s'effectue à partir de paires d'images "identiques" et "différentes", issues de la même catégorie générique d'objets (des voitures, des visages, ...).

Notre approche est aussi inspirée par les récents travaux de Moosmann [64]. De nombreux composants expliquent sa bonne performance en catégorisation d'image. Tout d'abord, un modèle de représentation par *sac-de-mots* rend la représentation robuste aux occultations et à de nombreuses transformations d'images (point de vue, illumination, ...). Deuxièmement, l'utilisation d'un ensemble d'arbres extrêmement aléatoires rend l'utilisation rapide (pour l'apprentissage et la validation) et permet de traiter des données de haute dimension (régions locales d'images).

Cette section est organisée comme suit. La section 5.3 présente notre approche, et la section 5.4 présente des résultats expérimentaux. Nous comparons nos résultats avec les méthodes de l'état de l'art dans la partie 5.4.4.

5.3 Mesure de similarité basée sur des parties locales

Comme nous l'avons déjà dit, notre intention est de définir une mesure de similarité pour prédire si deux images représentent le même objet ou non, malgré des modifications de point de vue, d'illumination, et la présence d'occultations (voir figure 5.2). Cette mesure doit donner de bonnes performances même si les objets en question n'ont jamais été vus auparavant. De plus, le système est entraîné à partir de paires d'images labélisées "identiques" ou "différentes", sans que la catégorie des images soit connue.

5.3.1 Quantification des différences locales

De même que dans les travaux de Ferencz et al [25], nous définissons la similarité de deux images à partir de l'échantillonnage de paires de régions locales correspondantes issues de ces deux images. Mais contrairement à Ferencz, nous ne limitons pas notre observation à la distance entre les descripteurs des régions locales, nous voulons aussi décrire la manière dont les régions locales diffèrent. Nous allons donc *caractériser les différences* d'apparence entre des régions locales correspondantes. Nous utilisons pour cela des arbres extrêmement aléatoires pour affecter une paire de régions locales à un cluster, qui représente une différence caractéristique entre ces régions locales.

Il aurait été possible de calculer un vocabulaire visuel à partir des régions locales des images de la base d'apprentissage [70] et de caractériser une paire de régions locales correspondantes par les deux mots de vocabulaire visuel les plus proches, et décider à partir de ces deux mots de vocabulaire visuel si les régions locales proviennent d'images identiques ou différentes. Cette approche souffre d'un grave défaut, comme nous le montrons dans la section 5.4.7: dès la première étape, elle supprime les différences fines entre les régions locales, car chaque région locale est affectée à un cluster, qui est une approximation de l'apparence de la région locale. Les différences fines ne sont pas importantes pour la classification de catégories d'objets (e.g. vélos, voitures, humains) mais elles jouent un rôle primordial dans la reconnaissance d'instances d'objets (le vélo de Pierre ou le vélo de Paul). De ce fait, nous proposons de (1) calculer des informations précises sur les régions locales (par exem-

ple comparer une dimension du descripteur SIFT à un seuil) et *ensuite* (2) de quantifier cette information avec des arbres extrêmement aléatoires. La première étape caractérise les différences fines, et la seconde réduit la complexité de l'espace de représentation. Cette section décrit le calcul des différences locales quantifiées, c'est à dire la construction du vocabulaire visuel des différences locales.

5.3.2 Vue d'ensemble

Le calcul de la mesure de similarité entre deux images est un processus en trois étapes, illustré figure 5.3. (a) Plusieurs paires de régions locales correspondantes sont échantillonnées dans la paire d'images. (b) Chaque paire de régions locales est quantifiée, c'est à dire qu'elle est affectée à un ensemble de clusters par des arbres extrêmement aléatoires. (c) La mesure de similarité globale entre les deux images est obtenue par combinaison linéaire des indicateurs d'appartenance aux différents clusters. Ces étapes sont détaillées ci-dessous.

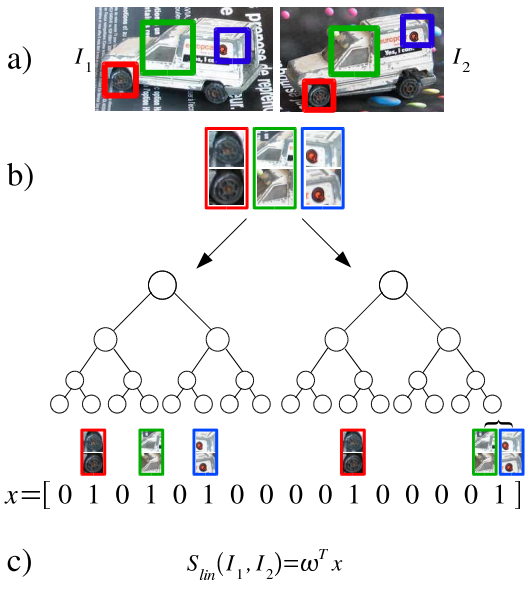


Figure 5.3: Calcul de similarités. (a) Détection de régions locales correspondantes. (b) Quantification de ces régions, i.e. affectation à des clusters par des arbres extrêmement aléatoires. (c) La mesure de similarité est une combinaison linéaire des indicateurs d'appartenance aux différents clusters.

5.3.3 Similarité entre deux images

Échantillonnage de paires de régions locales correspondantes

Chaque paire de régions locales est obtenue comme suit. Une région locale est échantillonnée aléatoirement dans la première image: la position (x,y) est choisie aléatoirement uniformément, et la taille de la région locale est choisie aléatoirement quadratiquement, ce qui favorise les plus petites régions locales (plus nombreuses et plus informatives [70]). Dans la seconde image, on choisit la région locale la plus similaire autour de la zone où la première région locale a été échantillonnée. La zone de recherche est centrée en (x,y), et sa taille est un multiple de la taille de la région locale. Cela permet d'avoir une zone de recherche plus grande que la région locale, et donc d'apporter une robustesse à de petites modifications d'échelle et d'orientation. La similarité est définie par corrélation croisée normée centrée (NCC). Le procédé est illustré figure 5.4.

Notons que notre algorithme suppose que les objets ont la même taille dans les deux images, ce qui permet de rechercher les correspondances à la même échelle dans les deux images. La section 5.4.2 montre que l'on peut travailler avec des objets de tailles différentes, ce qui nécessite de faire une recherche multi-échelles de régions locales correspondantes.

Quantification de l'espace des paires de régions locales

Chaque paire de régions locales est affectée à un ensemble de clusters par un ensemble d'arbres extrêmement aléatoires. La construction de ces arbres est détaillée section 5.3.4. Chaque paire de régions locales traverse l'ensemble des arbres, de la racine à une feuille, à chaque noeud de l'arbre le fils droit ou gauche est choisi en fonction de l'évaluation d'une condition simple sur la paire de régions locales. Quand une paire de régions locales atteint une feuille, cette feuille prend la valeur 1. Si une feuille n'est jamais atteinte par aucune paire de région locale, elle est mise à 0. On obtient donc un vecteur qui a pour dimension

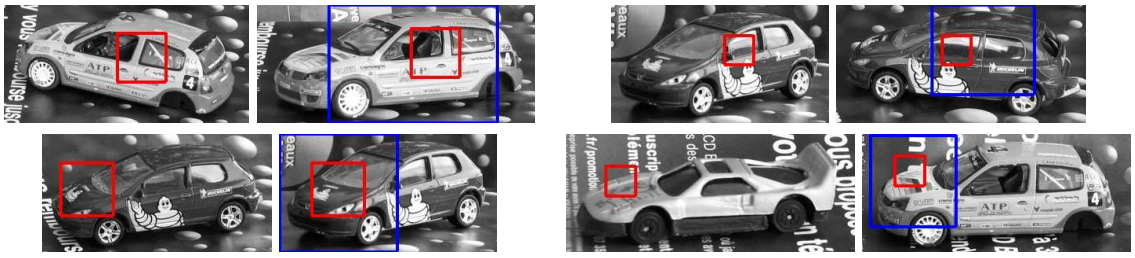


Figure 5.4: Paires de régions locales correspondantes échantillonnées dans la base des voitures jouets. Chaque paire d'images montre: une région locale dans l'image 1, la zone de recherche dans l'image 2 et la région locale la plus similaire trouvée dans la zone de recherche. Toutes les paires d'images sont positives (objets "identiques") hormis la dernière (objets "différents").

le nombre de feuilles dans la forêt, et qui prend les valeurs 1 et 0 indiquant si les feuilles correspondantes ont été atteintes ou non. Ce vecteur x est le descripteur global de la paire d'images. La représentation binaire (plutôt qu'un comptage par exemple) a été suggérée par [64].

Mesure de similarité

Les arbres appris séparent parfaitement les paires de régions locales sur lesquelles ils ont été entraînés, car ils ont justement été construits pour cela. Cependant, nous n'utilisons pas les arbres en tant que classifieurs, mais en tant que quantificateurs. Quand une paire de régions locales atteint une feuille, on ne s'intéresse donc pas à la prédiction de la feuille ("identique" ou "différent"), mais seulement à l'index de la feuille dans la forêt, car c'est celui-ci qui permet de mettre à jour le vecteur de description globale x de la paire d'images.

La mesure de similarité des images I_1 et I_2 est une simple combinaison linéaire des éléments de ce vecteur de description globale x: $S_{lin}(I_1,I_2) = \omega^\top x$, où ω est un vecteur de poids, optimisés de telle sorte que des valeurs élevées de S_{lin} soient obtenues pour des objets similaires, et des valeurs faibles pour des objets différents. En pratique, ω est la normale de l'hyperplan séparateur d'un SVM linéaire entraîné avec les vecteurs de description globale de paires d'images identiques et différentes.

Ces poids pondèrent les différentes feuilles, i.e. les différents clusters. Cela permet de mettre l'accent sur des feuilles très discriminantes, et d'ignorer les feuilles peu informatives.

5.3.4 Apprentissage des arbres extrêmement aléatoires

Chaque arbre est appris individuellement, suivant la procédure suggérée par Geurts [35] issue du paradigme "Perturbe et Combine" [11]. Nous échantillonnons une grande quantité de paires de régions locales dans les images positives (labélisées "identiques") et négatives (labélisées "différentes"). Nous créons ensuite un arbre avec un noeud unique, la racine, qui contient ces paires de régions locales positives et négatives. Puis nous créons des noeuds fils récursivement, en séparant en deux un noeud et ses paires de régions locales associées: une condition booléenne est affectée au noeud, et deux noeuds fils sont crées, le fils gauche contient les paires de régions locales pour lesquelles la condition à évaluer est fausse, le fils droit contient les autres. Cette procédure est répétée tant que les noeuds fils créés sont mixtes, c'est à dire qu'ils contiennent des paires de régions locales positives et négatives.

La condition booléenne affectée à un noeud (détaillée dans la section suivante) est une fonction paramétrique évaluée sur une paire de régions locales. Pour construire chaque noeud, nous générons un petit ensemble de conditions booléennes, et nous affectons au noeud la condition qui donne le plus grand gain en information: $IG = H - (n_1 H_1 + n_2 H_2)/n$, où H (respectivement H_1, H_2) et n (respectivement n_1, n_2) sont l'entropie et le nombre de régions locales correspondantes du parent (respectivement du fils gauche, fils droit). L'entropie est minimum (0) quand un noeud ne contient que des paires de régions posi-

tives ou négatives, et elle est maximum (approximativement 0.69) quand un noeud contient la même quantité non nulle de régions locales positives et négatives.

Nous utilisons des arbres extrêmement aléatoires pour plusieurs raisons. Premièrement, leur apprentissage est rapide, car contrairement au boosting, nous ne cherchons pas les valeurs optimales d'un paramètre, mais seulement la meilleure valeur d'un paramètre parmi un petit ensemble créé aléatoirement. Deuxièmement, l'utilisation d'arbres extrêmement aléatoires réduit le risque de sur-apprentissage, car le côté aléatoire des arbres les rend moins corrélés. Enfin, Moosmann [64] a montré que ces arbres sont bien adaptés au clustering.

5.3.5 Des conditions booléennes multi-modales

Nous proposons et combinons plusieurs types de conditions booléennes. Les trois premiers sont basés sur les informations contenues dans les régions locales échantillonnées (niveaux de gris, les gradients ou descripteur SIFT), le quatrième est basé sur des informations géométriques.

Conditions booléennes sur les niveaux de gris

Étant donnés les patchs P_1 et P_2 en niveaux de gris de deux régions locales, normalisés à une taille standard (15x15), luminosité standard (moyenne=0) et contraste standard (variance=1), la condition booléenne est vraie si:

$$|P_1(i,j) - P_2(i,j)| < d \tag{5.1}$$

où i, j, d sont des paramètres. i, j sont les coordonnées du pixel observé, et d est un seuil auquel la différence des niveaux de gris doit être inférieure pour que la condition soit vraie.

Conditions booléennes sur le gradient

Étant donnés les normes N_1, N_2 et orientations O_1, O_2 du gradient des patchs P_1, P_2 en niveaux de gris, la condition est vraie si:

$$|O_1(i,j) - O_2(i,j)| < d_o \wedge |N_1(i,j) - N_2(i,j)| < d_n \tag{5.2}$$

où i, j, d_o, d_n sont des paramètres. i, j sont les coordonnées du pixel observé, d_o et d_n sont les seuils sur les différences d'orientation et de gradient.

Conditions booléennes sur les descripteurs SIFT

Étant donnés les descripteurs SIFT [56] S_1 et S_2 des deux régions locales, la condition booléenne est vraie si:

$$k(S_1(i) - d) > 0 \wedge k(S_2(i) - d) > 0 \tag{5.3}$$

où i, d, k sont des paramètres. i est la dimension du descripteur SIFT considérée, d est un seuil, et $k = 1$ ou $k = -1$ indique si la valeur mesurée doit être supérieure ou inférieure au seuil.

Conditions booléennes sur la géométrie

Si (x,y,s) indiquent la position et l'échelle de la région locale échantillonnée dans la première image de la paire, la condition booléenne est vraie si:

$$k_x(x-d_x) > 0 \wedge k_y(y-d_y) > 0 \wedge k_s(s-d_s) > 0 \tag{5.4}$$

où $d_x, d_y, d_s, k_x, k_y, k_s$ sont des paramètres. k_x, k_y, k_s valent 1 ou -1 et indiquent si les valeurs mesurées doivent être supérieures ou inférieures aux seuils d_x, d_y, d_s. Cette condition booléenne peut représenter des notions complexes, par exemple une grande région locale située dans le coin inférieur gauche d'une image peut être exprimée par:

$$-1*(x-0.25) > 0 \wedge 1*(y-0.75) > 0 \wedge 1*(s-0.5) > 0$$

Affectation d'une condition booléenne à un noeud d'un arbre

Pour affecter une condition booléenne à un noeud interne d'un arbre, on génère des conditions de chaque type: on tire un type aléatoirement, puis on tire aléatoirement les paramètres requis par ce type. Le gain en information est calculé pour chacune de ces conditions, et la plus performante est affectée au noeud.

5.4 Résultats expérimentaux

5.4.1 Bases de données

Nous évaluons notre mesure de similarité sur quatre bases de données aux propriétés différentes. Nous utilisons notre propre base de données de voitures jouets, et des bases publiques afin de comparer nos résultats à l'état de l'art. Dans chacune de ces bases de données, les objets d'intérêt occupent presque totalement les images, et les paires d'images sont marquées comme positives ("identiques") ou négatives ("différentes"). Les paires d'images sont divisées en deux groupes: un ensemble d'apprentissage et un ensemble de test. Évidemment, l'ensemble de test ne contient aucune image qui soit dans l'ensemble d'apprentissage, mais il ne contient pas non plus les mêmes objets de la base d'apprentissage. La mesure de similarité est évaluée sur des *objets jamais vus*, et pas seulement sur des images jamais vues. Les bases de données sont détaillées ci-dessous, et elles sont illustrées figure 5.5.

Base des voitures jouets.[1] Elle contient 255 images de 14 modèles de véhicules différents (voitures et camions). Les images sont caractérisées par des modifications d'illumination, de fond, et de point de vue. La base d'apprentissage contient 1185 (resp. 7330) paires d'images positives (resp. négatives) issues de 7 objets différents. La base de test contient 1044 (resp. 6337) paires d'images issues des 7 autres objets (différents de ceux d'apprentissage donc).

[1] http://lear.inrialpes.fr/people/nowak

Base de voitures de Ferencz [25]. Elle contient 2868 paires d'images d'apprentissage (180 positives, 2688 négatives) et la base de test contient 2860 paires d'images. Cette base est caractérisée par la présence de paires d'images négatives (objets différents) qui ont une apparence visuelle très similaire (modèles de voitures ne se différenciant que par les enjoliveurs par exemple).

Base de visages de Jain [42]. C'est un sous ensemble[2] de la base "Faces in the news" [6] qui contient 500 paires positives et 500 paires négatives de visages, et nous mesurons notre performance par validation croisée en 10 paquets, comme les auteurs. Cette base de données est fabriquée automatiquement à partir d'images issues des actualités, elle est donc caractérisée par une grande variabilité d'apparence, d'illumination, de pose et d'expression, ainsi que par des erreurs d'annotation (en raison du procédé d'annotation automatique).

Base COIL-100 [66]. Elle est utilisée par Fleuret et al dans [31], qui en font 10 ensembles de bases, chacun étant composé d'une base d'apprentissage de 1000 paires d'images positives et négatives (issues de 80 objets) et de 250 paires d'images positives et négatives (issues des 20 autres objets). Cette base est caractérisée par des orientations très différentes entre les deux objets d'une paire, et la présence de catégories très hétérogènes: boites, tomates, médicaments, bouteilles, ...

Dans toutes ces expérimentations, les images sont considérées en niveaux de gris, et non en couleurs. Toutes les bases ont des paires d'objets qui ont à peu près la même orientation, hormis la base COIL-100 dont les objets ont des orientations quelconques.

La performance de notre mesure de similarité est mesurée à l'aide de la précision lors du taux d'erreur égale de la courbe précision-rappel (Precision for Precision-Recall Equal Error Rate, P-EER-PR) sur la base de test. Pour chaque paire d'images de la base de test, un score de similarité S_{lin} est calculé. Un seuil t est défini pour prédire si les objets sont identiques ou non: $S_{lin} > t$ signifie que les objets sont identiques, $S_{lin} \leq t$ signifie que les objets sont différents. La courbe précision-rappel est obtenue en faisant varier ce seuil t.

5.4.2 Évaluation paramétrique

Paramètres principaux

Nous avons mesuré l'influence de tous les paramètres qui interviennent dans le calcul de similarité sur la base des voitures jouets. La figure 5.6 montre l'effet des paramètres principaux, les autres sont détaillés dans la section suivante. Chaque courbe indique la précision lors du taux d'erreur égale de la courbe précision-rappel (P-EER-PR) en fonction d'un paramètre. Les courbes donnent la P-EER-PR de notre mesure de similarité S_{lin} ainsi que la P-EER-PR d'une mesure de similarité plus simple S_{vote}, qui utilise les arbres en tant que classifieurs et non quantificateurs. S_{vote} compte simplement le nombre de paires de régions locales qui atteignent des feuilles d'arbres labélisées "identiques".

[2] Nous remercions Jain Vidit de nous avoir envoyé le sous ensemble précis qu'il a utilisé

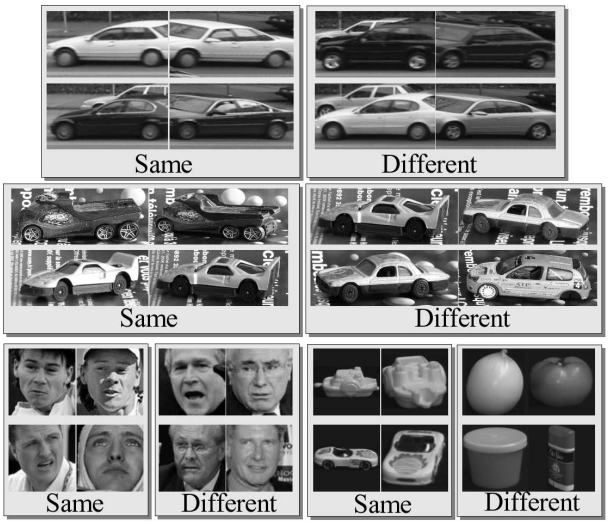


Figure 5.5: Deux paires d'images "identiques" et "différentes" sont présentées pour chaque base de données. Ligne 1: base de voitures de Ferencz. Ligne 2: notre base de voitures jouets. Ligne 3 gauche: Faces in the news. Ligne 4 droite: COIL-100. Bien que des images de paires "différentes" aient des apparences très similaires, et que des images de paires "identiques" aient des apparences très différentes, notre mesure de similarité obtient de très bonnes performances sur toutes ces bases (voir section 5.4.4).

Nous remarquons tout d'abord que la mesure de similarité S_{lin} obtient toujours une performance supérieure à la mesure de similarité plus simple S_{vote}, ce qui prouve que les arbres sont plus utiles en tant que quantificateurs plutôt qu'en tant que classifieurs. Deuxièmement, analysons l'influence des paramètres présentés figure 5.6. La première courbe montre que la zone de recherche de la région locale dans la seconde image de la paire ne doit pas être trop petite (sinon on ne peut trouver la région correspondante) ni trop grande (les chances de trouver un motif similaire mais issu d'une région différente augmentent). De plus, si la région de la seconde image est choisie totalement aléatoirement, la performance chute dramatiquement (42.7%, non indiqué sur le graphe). Cela montre à quel point le calcul de bonnes paires de régions locales correspondantes est crucial. La seconde courbe montre que plus il y a d'arbres dans la forêt, plus la performance augmente. C'est parce que nous obtenons plus de clusters, et que ces clusters sont décorrélés en raison du côté aléatoire de la construction des arbres. La troisième courbe montre que plus on échantillonne de paires de régions locales pour décrire une paire d'images, plus la performance augmente. Nous avons d'abord cru que c'était parce qu'en échantillonnant plus, on augmentait les chances d'échantillonner des régions très informatives. Si c'était le cas, seule S_{lin} progresserait, car celle ci fait la différence entre les informations discriminantes et non discriminantes (grâce à l'utilisation de la pondération par le SVM linéaire). Mais la mesure S_{vote} progresse elle aussi, et cette mesure ne fait aucune différence entre information très ou peu discriminante.

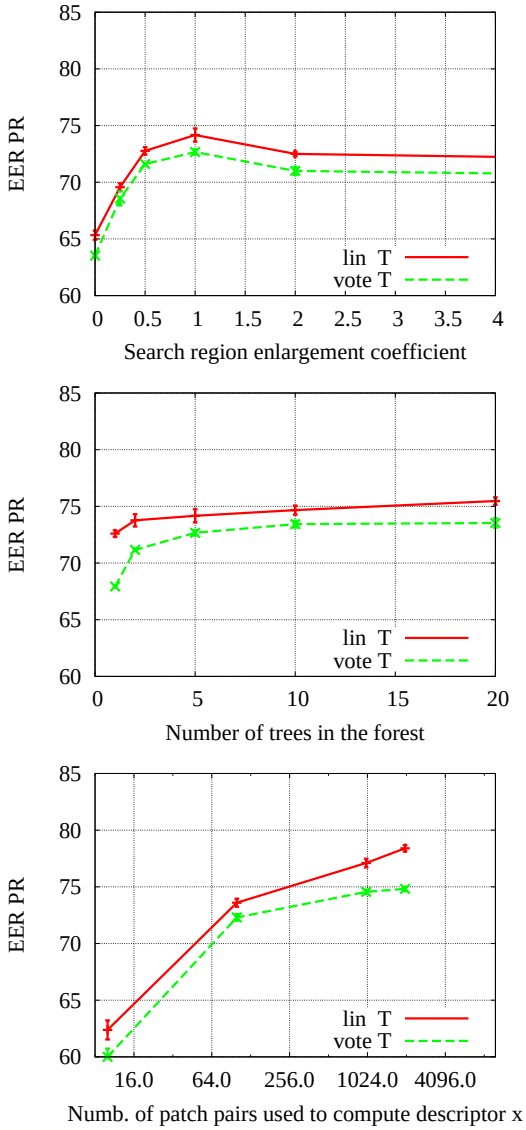


Figure 5.6: Précision lors du taux d'erreur égale de la courbe précision-rappel (P-EER-PR) sur la base de voitures jouets, avec une simple mesure de similarité S_{vote} et avec la mesure de similarité S_{lin} que nous proposons.

Cela signifie que toute information, même faible, améliore la performance, et confirme la conclusion de nos travaux précédents [70] sur les stratégies d'échantillonnage.

Paramètres secondaires

Nos expérimentations ont montré que les paramètres suivants ne sont pas des paramètres critiques du système: le nombre de paires de régions locales échantillonnées pour apprendre les arbres, les tailles minimum et maximum d'échantillonnage (figure 5.7), les paramètres de calcul des descripteurs SIFT (tableau 5.1).

Analysons tout d'abord la figure 5.7. Lors de ces expérimentations, la configuration standard est la suivante: 50 000 paires de régions locales sont échantillonnées pour apprendre les arbres, la meilleure condition booléenne est choisie parmi 500 conditions basées sur les niveaux de gris (équation 5.1), les forêts comportent 5 arbres, 150 paires de régions locales sont échantillonnées par paire d'images à décrire, d'une taille allant de 30 à 100 pixels de hauteur. On constate qu'il faut échantillonner une quantité minimum de régions locales pour apprendre les arbres, mais que l'échantillonnage d'une quantité plus grande ne diminue pas la performance, ce n'est donc pas un paramètre sensible. La taille maximale d'échantillonnage optimale est obtenue au maximum d'une courbe concave: si on ignore les régions de moyenne échelle, on se prive d'information, et si on considère des régions de trop grande échelle, l'information est trop globale. La taille minimum d'échantillonnage optimale est la taille de normalisation des vignettes, soit 15x15. En général, les petites échelles apportent beaucoup d'information dans les problèmes de classification, comme nous l'avons montré section 2.9. Cette section indique aussi que les tailles optimales d'échantillonnage dépendent des bases de données, et nous pensons que c'est le cas pour l'algorithme proposé ici aussi.

Analysons maintenant les paramètres de calcul de SIFT, résumés dans le tableau 5.1. La méthode de calcul standard des descripteurs SIFT proposée par David Lowe [56] donne la meilleure performance. L'algorithme standard applique une gaussienne de floutage sur les niveaux de gris afin d'atténuer le signal loin du centre pour compenser les imprécisions des détecteurs de régions d'intérêt. Cela n'a pas de sens dans notre cas car les régions sont choisies aléatoirement, nous avons donc supprimé cette gaussienne, mais cela ne change pas la performance. Utiliser une plus grande quantification spatiale (grille de 8x8 au lieu de 4x4) ou calculer le gradient sur une région sous-échantillonnée font perdre respectivement 1 et 2% de performance. De ce fait, nous utilisons simplement l'algorithme standard de calcul de descripteurs SIFT.

Expérimentation	P-EER-PR
SIFT standard	81.86±0.5
Gaussienne de floutage ignorée	81.82±0.7
Quantification spatiale de 8x8 plutôt que 4x4	80.96±0.9
Calcul du gradient sur une région plus petite	79.80±0.3

Table 5.1: Précision lors du taux d'erreur égale précision-rappel avec différentes méthodes de calcul du descripteur SIFT sur la base des voitures jouets

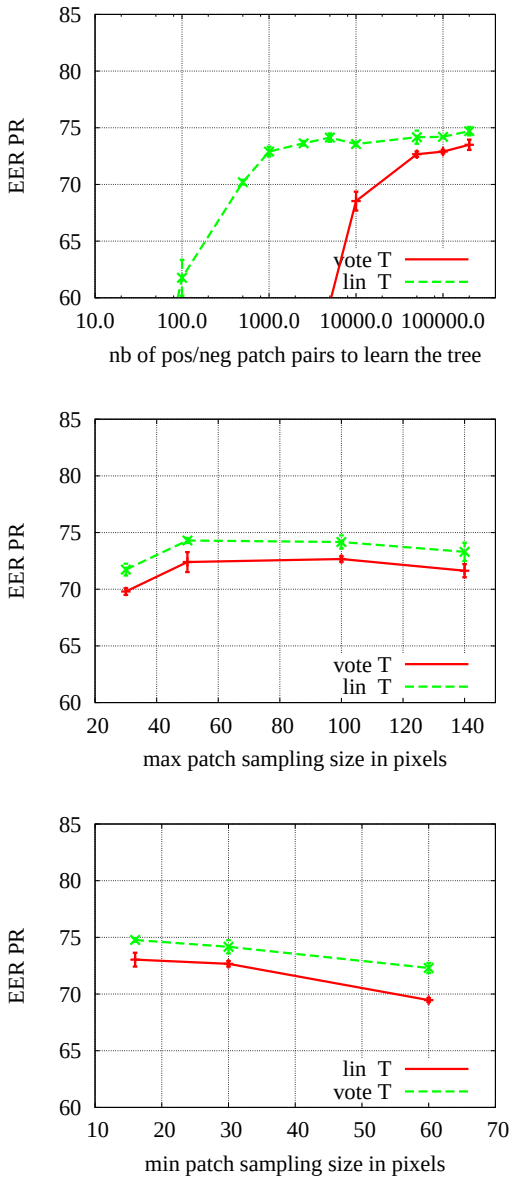


Figure 5.7: Influence du nombre de paires de régions locales échantillonnées pour apprendre un arbre (ligne 1) et de la taille maximum et minimum d'échantillonnage des régions locales dans les images (resp. lignes 2 et 3). Les courbes indiquent la précision lors du taux d'erreur égale de la courbe précision-rappel (P-EER-PR).

Enfin, nous avons étudié les paramètres influençant la recherche de la région locale dans la seconde image la plus similaire à la région locale de la première image. Nous avons déjà considéré la taille de la zone de recherche dans la section précédente.

Le tableau 5.2 montre l'intérêt de faire une recherche multi-échelles de la seconde région locale. L'algorithme standard recherche une région locale de taille identique dans la seconde image, dans un voisinage de la position d'échantillonnage de la première région locale. Le tableau montre qu'en augmentant le nombre d'échelles et le pas d'échelle, on peut obtenir une amélioration de performance jusqu'à +2%. La base de données utilisée est la base des voitures jouets. Passer d'une à cinq échelles fait gagner un point de performance. Passer de cinq à neuf échelles en utilisant un pas deux fois plus petit (donc les échelles minimale et maximale sont les mêmes, mais il y a deux fois plus d'échelles intermédiaires) fait gagner un autre point de performance. Malgré les bons résultats de la recherche multi-échelles, nous ne considérons que la recherche à une seule échelle dans le reste de ce document, en raison du temps de calcul prohibitif de la recherche multi-échelles.

Enfin, nous avons considéré divers espaces pour la recherche de la région locale la plus similaire dans la seconde image: l'espace SIFT et l'espace des niveaux de gris. Dans toutes les expérimentations de ce document, on recherche le plus proche élément dans l'espace des niveaux de gris, en effectuant une corrélation normée centrée (NCC). Comme l'arbre utilise des conditions booléennes à base de descripteurs SIFT, on peut imaginer que la recherche du plus proche descripteur dans l'espace SIFT représente un avantage. Le tableau 5.3 présente les résultats obtenus sur une sous-partie de la base de voitures jouets. Nous avons travaillé sur une sous-partie car le temps de calcul d'une grande quantité de descripteurs SIFT est prohibitif. La sous-base contient 300 paires positives et 600 paires négatives, et le modèle est appris avec 5 arbres, 50 000 paires de régions locales sont échantillonnées pour les apprendre, la meilleure condition booléenne basée sur des conditions SIFT est choisie parmi 500 conditions aléatoires, 500 paires de régions locales sont échantillonnées pour décrire une paire d'images, et les tailles d'échantillonnages varient entre 15 et 100 pixels de côté. On constate que la recherche du plus proche voisin dans l'espace des niveaux de gris est largement plus performante que la recherche dans l'espace SIFT. Ce résultat nous a surpris, mais la robustesse et la tolérance à de petites variations (échelle, orientation, niveaux de gris) du descripteur SIFT le rend trop tolérant pour rechercher l'élément le plus similaire, ce que confirment les travaux de Jain Vidit et al [42] et Mark Everingham et al [21].

Nombre d'échelles	Pas d'échelle	P-EER-PR
9	0.9	83.08±0.6
5	0.9	82.76±0.5
5	0.81	82.40±0.3
1	-	81.42±0.4

Table 5.2: Étude des paramètres du matching pour le calcul des paires de régions locales. Précision lors du taux d'erreur égale de la courbe précision-rappel en fonction des paramètres de recherche multi-échelles.

Espace	P-EER-PR
SIFT	60%
Niveau de gris	70%

Table 5.3: Recherche de la région locale dans la seconde image la plus similaire à celle issue de la première image dans l'espace SIFT ou dans l'espace des niveaux de gris. Précision lors du taux d'erreur égale de la courbe précision-rappel (P-EER-PR).

5.4.3 Connaissances génériques et spécifiques

Notre algorithme apprend deux types d'information à partir des données d'apprentissage: les arbres, et les poids qui viennent pondérer les différents clusters. On peut se demander à quel point les informations apprises sont *spécifiques à une catégorie*. L'algorithme apprend-il des règles de calcul de similarités très générales applicables à toutes les bases, ou bien apprend il des règles spécifiques à la base de données sur laquelle il est entraîné? Des éléments de réponse sont présentés dans le tableau 5.4. Il indique, pour deux bases de données, combien la P-EER-PR diminue quand d'autres bases sont utilisées pour l'apprentissage des arbres et des poids (T+W), ou des arbres seulement (T), auquel cas les poids sont appris sur la base d'apprentissage associée à la base de test.

Premièrement, on observe que la meilleure performance est atteinte quand l'apprentissage et le test se font sur la même base d'images, ce qui prouve que l'algorithme apprend des connaissances spécifiques. Deuxièmement, apprendre les arbres et les poids sur une autre base est moins bon qu'apprendre les arbres sur une autre base et les poids sur la bonne base. Cela signifie que le quantificateur joue un rôle moins important que la pondération, et qu'à priori tout quantificateur peut être utilisé, même si celui qui donne la meilleure performance est celui appris sur la bonne base. Troisièmement, il est plus important d'apprendre des arbres sur la bonne base pour la base Ferencz que pour la base COIL-100. C'est parce que les images de la base COIL-100 ont différentes orientations et représentent des objets très hétérogènes, alors que la base Ferencz contient des images alignées de véhicules vus de profil, ce qui facilite grandement l'apprentissage d'informations spécifiques (et utiles).

	Toy cars		Ferencz		Visages		COIL-100	
	T+W	T	T+W	T	T+W	T	T+W	T
Ferencz	28.0	4.5	0	0	49.2	23.5	35.8	10.1
COIL-100	10.0	1.6	9.0	2.4	13.2	2.8	0	0

Table 5.4: Les arbres (T) et les poids (W) sont appris par notre algorithme. Ce tableau montre le nombre de points de P-EER-PR perdus sur les bases Ferencz et COIL-100 quand les arbres (T) ou les arbres et les poids (T+W) sont appris sur une autre base.

Le tableau 5.5 donne plus de détails sur cet apprentissage croisé. Les résultats du paragraphe précédent sont issus de ce tableau plus complexe. Les paramètres expérimentaux sont les suivants: 50 000 paires de régions locales sont échantillonnées pour apprendre 5 arbres, la meilleure condition booléenne basée sur les descripteurs SIFT est choisie parmi 500, 250 paires de régions locales sont échantillonnées pour décrire une paire d'images. Ce tableau donne la performance (P-EER-PR) avec apprentissage de certains éléments sur une base, et évaluation sur une autre. Dans le tableau du haut, la forêt et le SVM sont appris sur la base spécifiée en ligne et utilisés sur la base spécifiée en colonne. On constate clairement qu'il faut utiliser la base d'origine pour obtenir la meilleure performance. Dans le tableau du bas, la forêt est apprise sur la base spécifiée en ligne, mais le SVM est appris sur les paires d'apprentissage de la base spécifiée en colonne. En général, la meilleure performance est obtenue en apprenant l'arbre sur la base utilisée pour l'évaluation. Mais d'autres bases permettent d'obtenir des résultats très proches (Voiture jouets pour utilisation sur Ferencz, perte de 4%; autres bases pour utilisation sur COIL-100, perte de 1% et plus; COIL-100 pour utilisation sur Visages, perte de 3%) et parfois même supérieurs (COIL-100 pour utilisation sur Voitures jouets, gain de moins de 1%). L'utilisation de forêts totalement aléatoires, c'est à dire dont les conditions sont affectées aléatoirement aux noeuds sans considération des données, donne un score assez mauvais sur les bases spécifiques (Voitures jouets et Ferencz) et un score correct sur les bases génériques (COIL-100 et Visages).

Cela confirme les conclusions du paragraphe précédent: la forêt sert principalement à quantifier les différences, et le SVM permet de mettre l'accent sur les différences fondamentales. Les mauvais résultats des expérimentations croisées du tableau du haut confirment que le rôle du SVM est crucial. Le tableau du bas indique que la quantification des différences n'est pas critique, mais que la performance est en général meilleure si la quantification est adaptée aux données traitée, en utilisant la bonne base pour apprendre la forêt.

5.4.4 Performance et comparaison avec l'état de l'art

Cette section présente la performance de notre algorithme sur différentes bases, et la comparaison avec les résultats publiés dans l'état de l'art. Pour chaque base de données, le protocole expérimental des méthodes comparées a été scrupuleusement respecté. Les résultats sont présentés dans le tableau 5.6.

Le même jeu de paramètres a été utilisé pour toutes les bases de données. 10 000 paires de régions locales sont échantillonnées pour apprendre les arbres, pour chaque noeud la meilleure condition booléenne est choisie parmi un ensemble de 1000 conditions créées aléatoirement, les forêts contiennent 50 arbres, 1000 paires de régions locales sont échantillonnées dans une paire d'images pour la décrire, la zone de recherche d'une région locale dans la seconde image est augmentée d'une fois la taille de la région locale dans les quatre directions (ce qui produit une région 9 fois plus grande), la taille minimum d'échantillonnage des régions locales est 15x15 pixels, et la taille maximum est la moitié de la hauteur des images. En moyenne, les arbres produits ont 20 niveaux.

Apprentissage	Voitures jouets	Ferencz	COIL-100	Visages
Voitures jouets	81.4	54.7	80.8	66.0
Ferencz	72.5	82.7	81.6	70.0
COIL-100	73.7	46.9	90.8	64.0
Visages	66.9	33.5	77.6	81.0

Apprentissage	Voitures jouets	Ferencz	COIL-100	Visages
Voitures jouets	81.4	78.2	89.2	72.0
Ferencz	79.8	82.7	88.4	72.0
COIL-100	82.1	72.6	90.8	78.0
Visages	79.4	59.2	88.0	81.0
Forêt aléatoire	$69.5_{\pm 1.3}$	$44.3_{\pm 4.1}$	$84.5_{\pm 1.3}$	$79.2_{\pm 1.0}$ x

Table 5.5: Le modèle est appris sur une base et évaluée sur une autre. Les tableaux reportent la P-EER-PR. Tableau du haut: la forêt et le SVM sont appris sur la base spécifiée en ligne et utilisés sur la base spécifiée en colonne. Tableau du bas: la forêt est apprise sur la base spécifiée en ligne, mais le SVM est appris sur les paires d'apprentissage de la base spécifiée en colonne; pour la dernière ligne la forêt n'est pas apprise avec une base mais créée totalement aléatoirement.

Méthode	Voitures jouets	Ferencz	Visages	COIL-100
État de l'art	-	84.9 [24]	70.0 [42]	$88.6_{\pm 4}$ [31]
Méthode proposée	$85.9_{\pm 0.4}$	$91.0_{\pm 0.6}$	$84.2_{\pm 3.1}$	$93.0_{\pm 1.9}$
Gain	-	6.1	14.2	4.4

Table 5.6: Précision lors du taux d'erreur égale de la courbe précision-rappel (P-EER-PR) sur différentes bases de données. La méthode proposée surpasse clairement les autres méthodes.

Base de voitures jouets. Sur notre base de voitures jouets, nous obtenons la précision lors du taux d'erreur égale pour la courbe de précision-rappel (P-EER-PR) de 85.9%$\pm$0.4, moyennes et écart-type calculés sur 5 executions. L'utilisation d'une simple corrélation normée centrée (NCC) comme mesure de similarité donne une P-EER-PR de 51.1%. Cette base de données est nouvelle, il n'y a pas d'autres résultats publiés auxquels se comparer.

Base de voitures de Ferencz. Sur cette base [24], nous obtenons une P-EER-PR de

Méthode	Rappel 40%	Rappel 60%	Rappel 80%
Jain [42]	$93.0_{\pm 6.3}$	$78.9_{\pm 8.2}$	$60.1_{\pm 7.0}$
Méthode proposée	$99.0_{\pm 1.9}$	$97.8_{\pm 2.5}$	$86.3_{\pm 7.6}$

Table 5.7: Précision sur la base de Jain pour des valeurs données de rappel.

91.0%±0.6, quand Ferencz obtient 84.9%. La figure 5.8 montre des paires d'images classées par notre mesure de similarité.

Base de visages de Jain. Jain [42] obtient une performance bien meilleure que les algorithmes ayant obtenu la meilleure performance sur la base de référence FERET, et nous obtenons une performance bien meilleure que la sienne. La P-EER-PR présenté dans le tableau 5.6 (70%) est approximatif car il est estimé à partir d'un graphe issu de leur article, nous présentons donc aussi des performances avec la même métrique que les auteurs, la précision pour taux de rappel donné, dans le tableau 5.7. Nous obtenons toujours une performance bien meilleure, allant même jusqu'à un gain de 26% pour un taux de rappel de 80%. De plus, Jain utilise des images de visage réctifiées à une orientation standard, alors que nous utilisons les images d'origine. La figure 5.9 présente des paires d'images classées par notre mesure de similarité.

Base COIL-100. Sur la base COIL-100, nous avons un taux de P-EER-PR de 93.0%±1.9 quand Fleuret [31] obtient 88.6%±4. De plus, l'algorithme utilisé par Fleuret nécessite les labels individuels des différents objets lors de l'apprentissage, alors que nous nous contentons de contraintes d'équivalences (objets "identiques" ou "différents").

5.4.5 Combinaison des types de condition booléenne

Nous avons comparé et combiné différents types de conditions booléennes, basées sur les niveaux de gris (équation 5.1), le gradient (équation 5.2), le descripteur SIFT (équation 5.3)

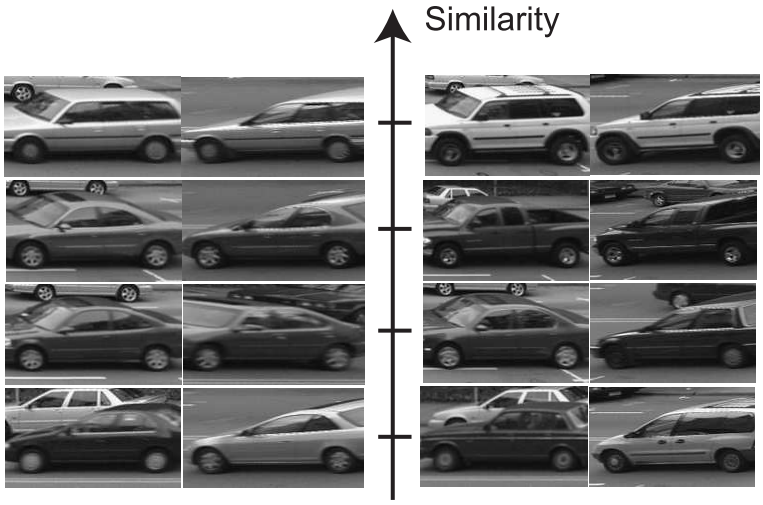


Figure 5.8: Paires d'images issues de la base de test Ferencz, ligne 1: les deux paires "identiques" les plus similaires, ligne 2: les deux paires "identiques" les moins similaires, ligne 3: les deux paires "différentes" les moins différentes, ligne 4: les deux paires "différentes" les plus différentes (selon la mesure de similarité apprise).

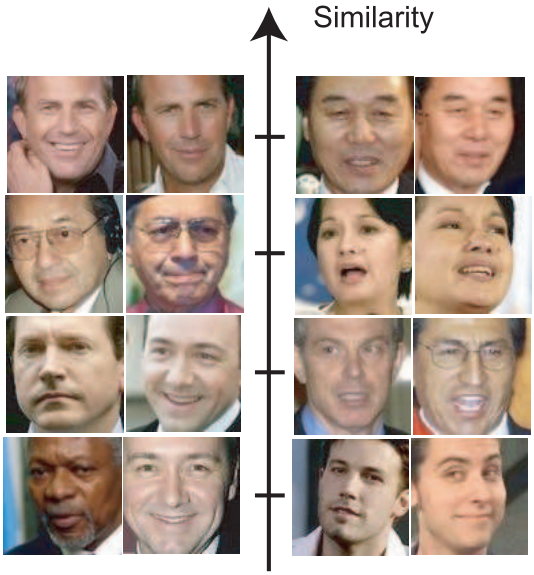
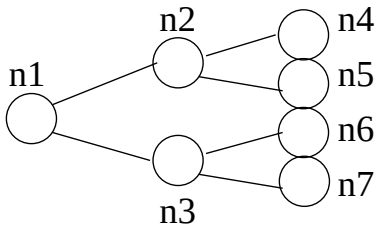


Figure 5.9: Paires d'images issues de la base de test de visages, ligne 1: deux paires très similaires, ligne 2: deux paires similaires, ligne 3: deux paires différentes, ligne 4: deux paires très différentes.



Noeud	#pos	#neg	Entropie	Condition booléenne
n1:	10000+	10000-	0.69314	|P1(2,3)-P2(2,3)|<0.72
n2:	174+	317-	0.65011	|P1(9,8)-P2(9,8)|<0.48
n3:	9826+	9683-	0.69312	|P1(2,2)-P2(2,2)|<0.31
n4:	19+	60-	0.55166	|P1(3,4)-P2(3,4)|<0.22
n5:	155+	257-	0.66218	|P1(9,2)-P2(9,2)|<0.15
n6:	1845+	3040-	0.66292	|P1(8,5)-P2(8,5)|<0.32
n7:	7981+	6643-	0.68895	|P1(3,3)-P2(3,3)|<0.93

Figure 5.10: Les trois premiers niveaux d'un arbre calculé sur des données réelles. Pour chaque noeud, on indique combien de paires de régions locales positives ("identiques") et négatives ("différentes") il contient, l'entropie associée et la condition booléenne affectée au noeud. Ici, les conditions booléennes utilisent des niveaux de gris (équation 5.1).

Condition(s) booléenne(s)	P-EER-PR
Niveaux de gris (équation 5.1)	77.2
Gradient (équation 5.2)	78.3
SIFT (équation 5.3)	81.9
SIFT (5.3) + Géométrie (5.4)	83.1
Tout (5.1+5.2+5.3+5.4)	79.3
Chance	14

Table 5.8: P-EER-PR en fonctions des différentes conditions booléennes utilisées dans les arbres.

et la géométrie (équation 5.4). Dans cette section, nous évaluons la performance d'arbres qui utilisent une ou plusieurs conditions booléennes. Les trois premiers niveaux d'un arbre utilisant exclusivement des conditions basées sur les niveaux de gris sont présentés figure 5.10.

Les résultats sur les performances individuelles et les combinaisons sont présentés dans le tableau 5.8, ils sont mesurés sur la base des voitures jouets. On constate que les conditions portant sur les descripteur SIFT sont plus performantes que les conditions portant sur les niveaux de gris et les gradients (gain de 4.7 et 3.1%). Cela s'explique par la robustesse du descripteur SIFT, et à sa tolérance à de faibles variations d'échelle, de position et de rotation. On constate aussi que la combinaison de SIFT et de la géométrie apporte un gain significatif par rapport à SIFT seul: 1.2%. Le tableau 5.9 donne les performances détaillées sur chaque base en utilisant des conditions booléennes basées sur SIFT combinées ou non avec des conditions géométriques, et il indique que sur les différentes bases la géométrie permet de gagner environ 1%, que la forêt contienne 5 ou 50 arbres. Par contre, la combinaison de tous les critères (79.3%) est moins performante que l'utilisation du meilleur de ces critères seul (SIFT, 81.9%). L'explication la plus probable est qu'il est plus simple de faire du sur-apprentissage avec des conditions sur les niveaux de gris que sur SIFT, et donc que les conditions sur les niveaux de gris sont préférées aux conditions sur SIFT lors de la création de l'arbre, ce qui n'est pas une stratégie payante sur les données de test.

Géométrie	Nombre d'arbres	Toy cars	Ferencz	Visages	COIL-100
oui	50	85.84±0.4	91.02±0.6	84.2±3.5	93.0±1.8
oui	5	84.64±0.7	90.10±0.6	84.2±3.2	92.1±1.5
non	50	84.65±0.3	89.22±0.8	85.0±3.4	93.6±1.8
non	5	83.60±0.3	87.88±0.4	83.6±3.6	92.0±2.1

Table 5.9: Performance (P-EER-PR) sur les quatre bases en fonction de l'utilisation de conditions géométriques et du nombre d'arbres.

5.4.6 Visualisation des similarités

Dans cette section nous visualisons les similarités de manière détaillée sur deux bases de données: la base des voitures jouets et la base infra-rouge Robin.

Base des voitures jouets

La section précédente a démontré que notre mesure de similarité surpasse les mesures proposées dans l'état de l'art. Comme "une image vaut mieux qu'un long discours", nous proposons de visualiser en deux dimensions la mesure de similarité évaluée sur les paires d'images de la base des voitures jouets. Ces images proviennent de l'ensemble de test, les modèles de véhicules n'ont donc jamais été vus lors de l'apprentissage.

Pour ce faire, nous calculons une projection 2D des images en utilisant la mesure de similarité proposée. Cette projection se fait par 2D multidimensional scaling (MDS). Cela consiste à calculer une projection 2D telle que les distances deux à deux entre exemples projetés reflète le mieux possible les similarités prédites calculées deux à deux. Cette projection est présentée figure 5.11, en bas. Il est surprenant d'observer à quel point les différents véhicules sont bien séparés, même pour des véhicules très similaires (en bas et en haut à gauche par exemple). A titre de comparaison, la figure 5.11, en haut, présente une projection 2D basée sur une autre mesure de similarité: la distance Chi2 entre des descripteurs issus d'une représentation d'images par sac-de-mots. Cette comparaison montre clairement la supériorité de notre mesure de similarité.

Les figures 5.12, 5.13, 5.14 et 5.15 présentent des paires d'images de la base de voitures jouets et la mesure de similarité prédite par notre algorithme. Le seuil de décision identique/différent choisi est celui qui maximise la P-EER-PR, avec un seuil nous pouvons donc passer d'un score de similarité à une décision: identique ou différent. Les quatre figures présentent respectivement des faux positifs, des faux négatifs, des vrais négatifs et des faux positifs. On observe visuellement que la cause d'erreur principale pour les faux positifs est une grande similarité visuelle, et la cause d'erreur principale pour les faux négatifs est une grande différence dans l'orientation des véhicules. Les figures 5.16 et 5.17 présentent des cas difficiles que l'algorithme est parvenu à classer correctement, avec respectivement une grande variabilité dans l'orientation et l'illumination.

Base Robin

Dans cette section, nous faisons une analyse visuelle des performances sur la base infra-rouge Robin[3]. Cette base de données contient des véhicules civiles fimés par caméra infra-rouge. Nous utilisons environ 400 images, à partir desquelles nous formons des paires

[3]http://robin.inrialpes.fr, base de données numéro 1: Détection multiclasses avec changement d'orientation, produite par Cybernetix et Bertin Technologies pour le projet Technovision

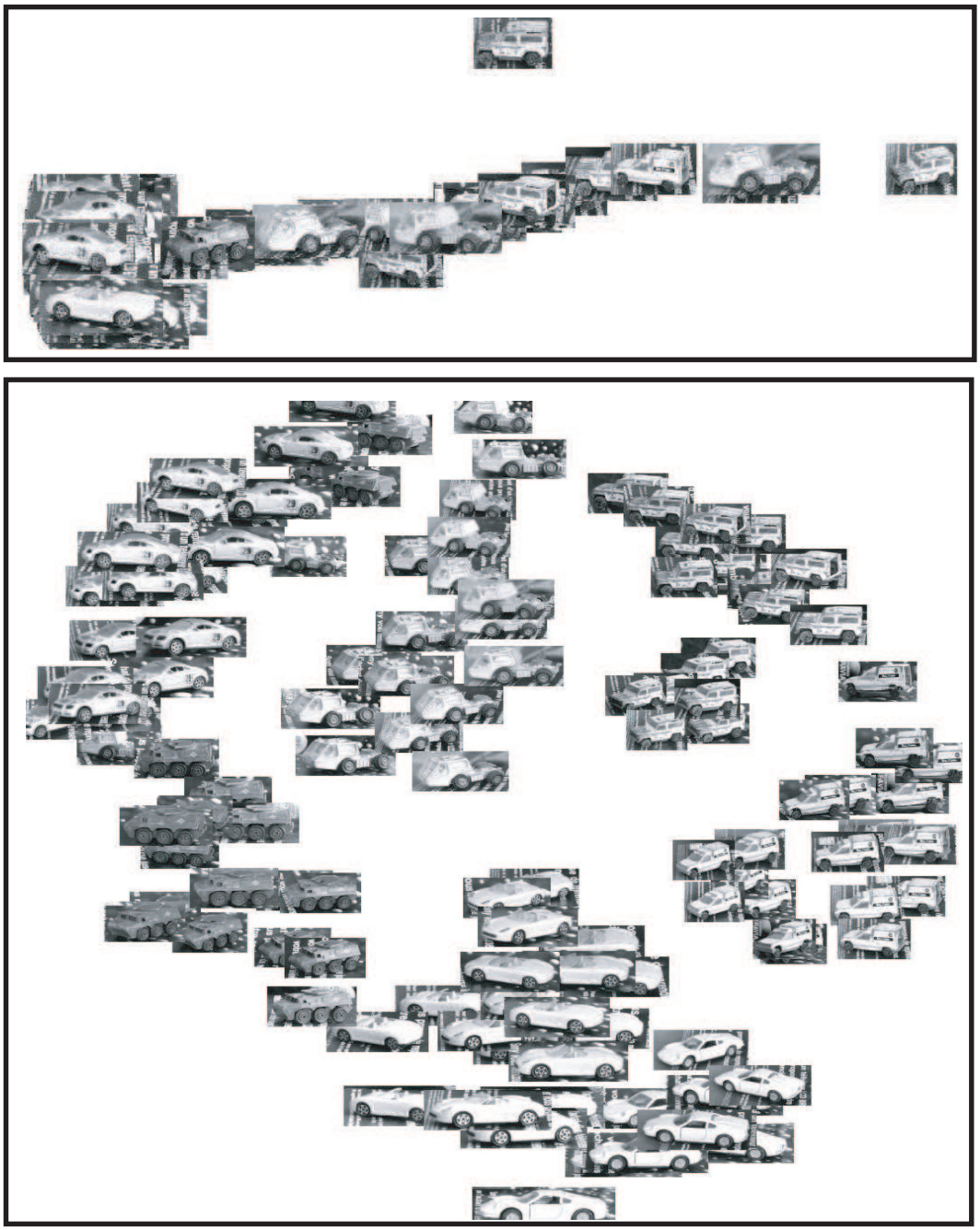


Figure 5.11: Représentation 2D de la mesure de similarité calculée sur des paires d'images de voitures, dont les modèles n'ont jamais été vus lors de l'apprentissage. Haut: Similarité calculée par une représentation sac-de-mots et une distance Chi2. Bas: Similarité calculée par la mesure proposée, qui regroupe très bien les différentes vues du même objet.

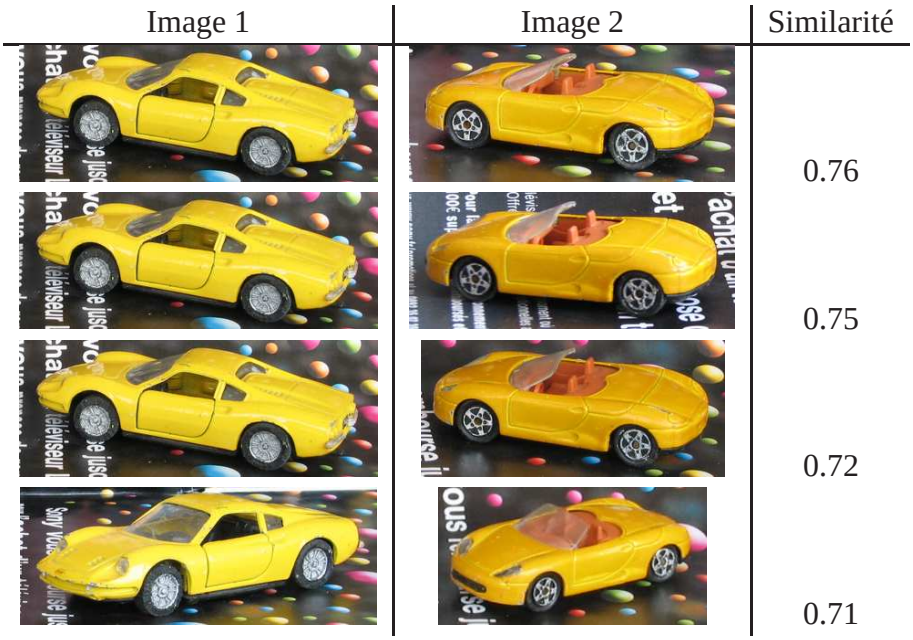


Figure 5.12: Faux positifs: Modèles différents prédits comme similaires, en raison d'une grande similarité visuelle

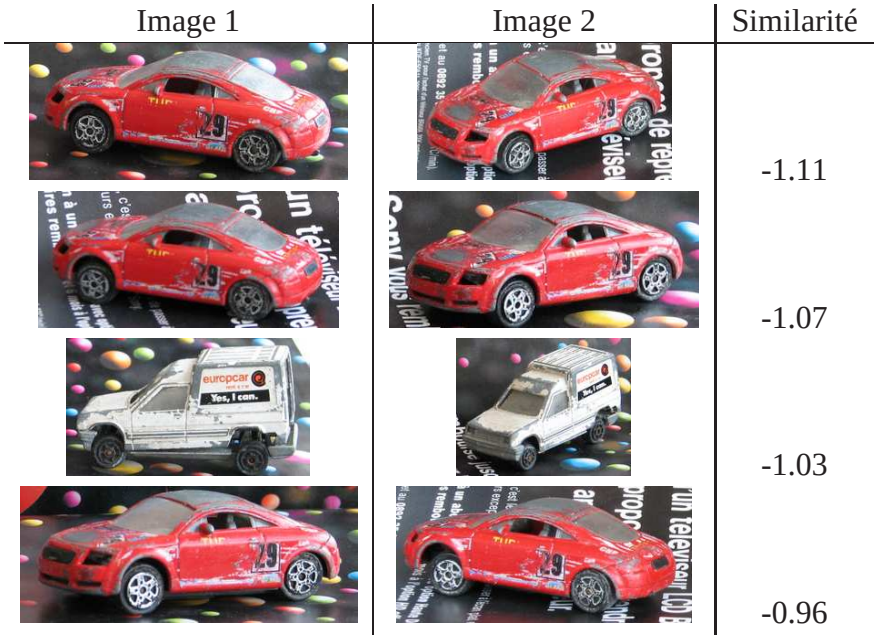


Figure 5.13: Faux négatifs: Modèles similaires prédits comme différents, en raison d'une grande différence d'orientation

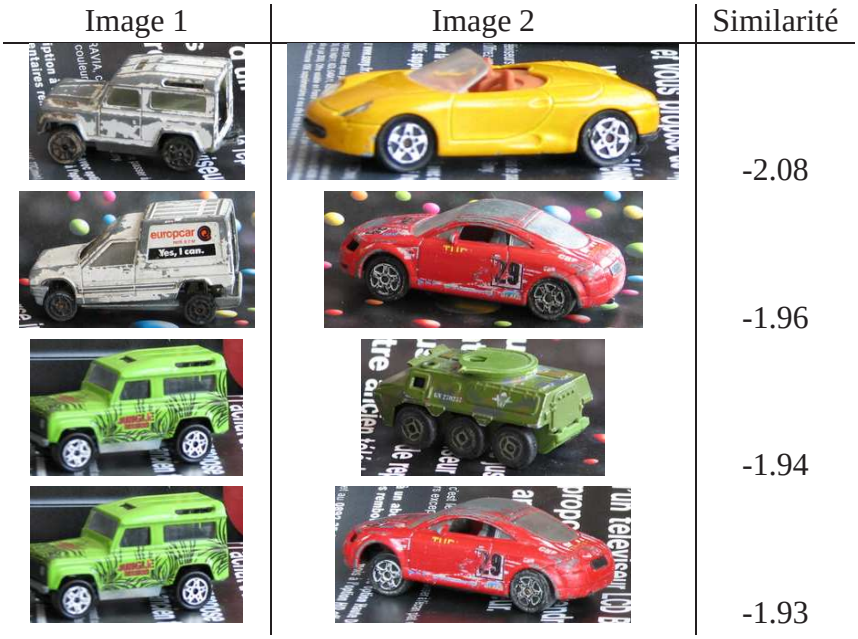


Figure 5.14: Vrais négatifs: Modèles différents prédits comme tels par notre mesure

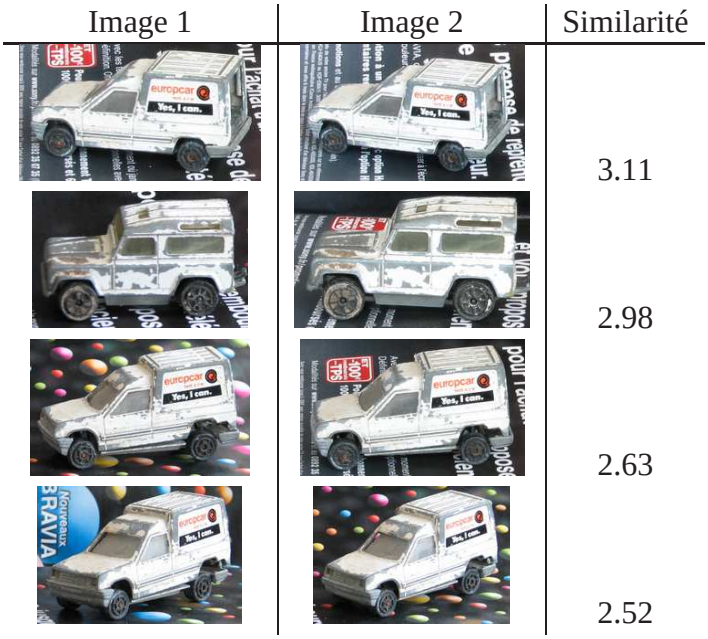


Figure 5.15: Vrais positifs: Modèles similaires prédits comme tels par notre mesure

positives et des paires négatives. De telles paires peuvent être observées figures 5.18, 5.19, 5.20, 5.21.

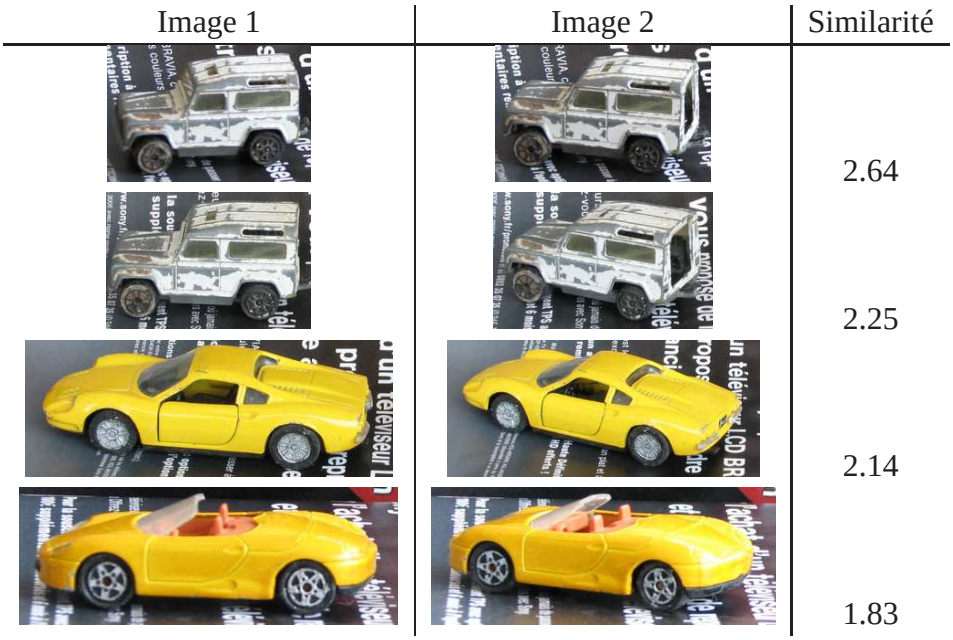


Figure 5.16: Vrais positifs: Modèles similaires prédits comme tels par notre mesure, malgré des changements d'orientation.

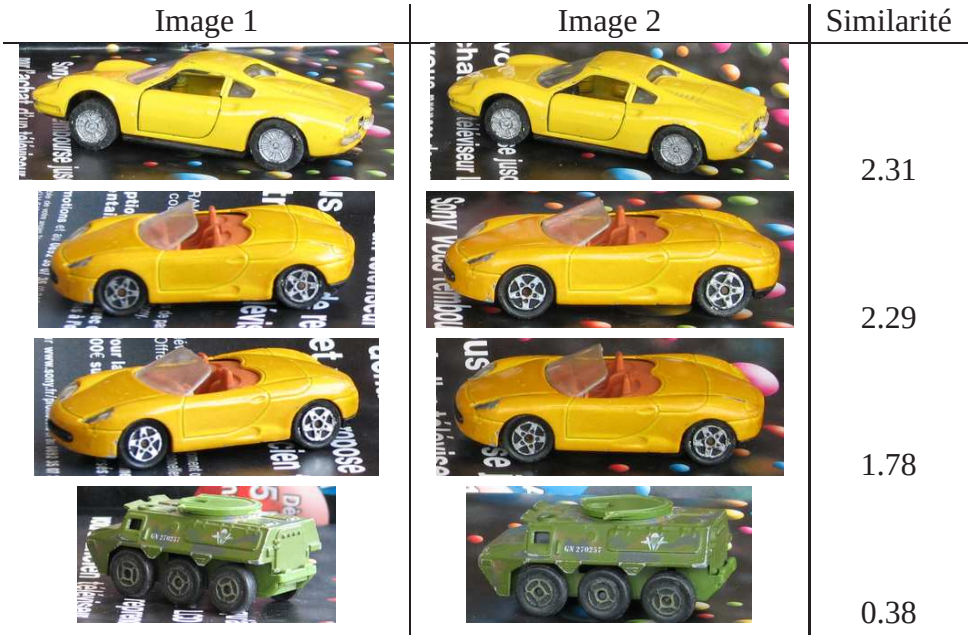


Figure 5.17: Vrais positifs: Modèles similaires prédits comme tels par notre mesure, malgré des changements d'illumination.

La figure 5.18 compare la prédiction de similarité sur des modèles de voitures inconnus avec une mesure par corrélation en niveaux de gris et la mesure proposée. On constate que la mesure proposée regroupe mieux les paires similaires (zone bleue à droite) que la corrélation. De plus, la valeur du score de confiance est plus prononcée pour les exemples très similaires (courbe rouge, partie droite).

Si l'algorithme était parfait, toutes les paires "Similaires" de test (bleu) seraient à droite, et toutes les paires "Différentes" de test (marron) seraient à gauche. Un algorithme bon mais imparfait placerait beaucoup de paires "Similaires" à droite, beaucoup de paires "Différentes" à gauche, et se tromperait au milieu, hors de la zone de confiance (score très élevé ou très faible). Ici, on remarque que l'algorithme proposé place des paires "Différentes" dans la zone de confiance des paires "Similaires", et vice-versa. L'analyse des erreurs, figure 5.19, nous permet de comprendre pourquoi. On constate que les paires différentes ont une apparence très similaires, et que les paires similaires ont une apparence très différente. Et nous rappelons qu'il s'agit de véhicules jamais vus lors de l'apprentissage. Les similarités visuelles apprises sur les véhicules d'apprentissage ne permettent pas de généraliser suffisamment pour résoudre ces cas difficiles.

La figure 5.20 présente des paires d'images différentes reconnues comme telles. La figure 5.21 présente des paires similaires reconnues comme telles. Parmi ces paires similaires, on observe des paires d'apparence visuelle très similaires, mais aussi des paires où l'orientation varie, avec présence d'occultations ou de clutter (faible).

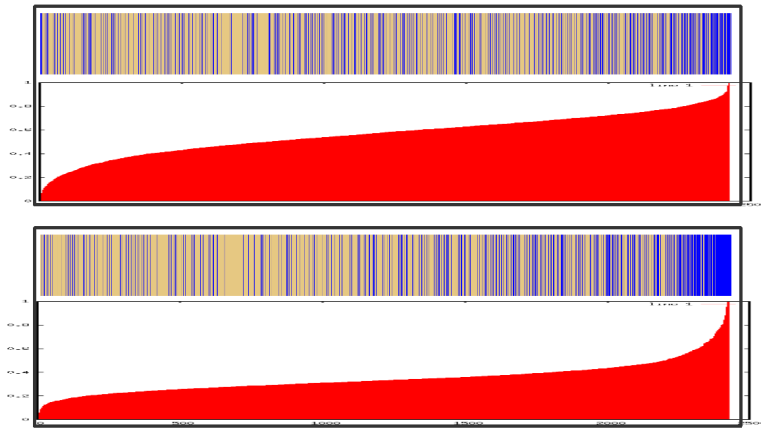


Figure 5.18: Base Robin: Comparaison d'une similarité calculée par corrélation en niveau de gris (haut) ou par la mesure proposée (bas). Les paires d'images positives (trait bleu) et négatives (trait marron) sont classées par similarité prédite croissante, indiquée par la courbe rouge. La mesure proposée se comporte significativement mieux que la corrélation en niveaux de gris.

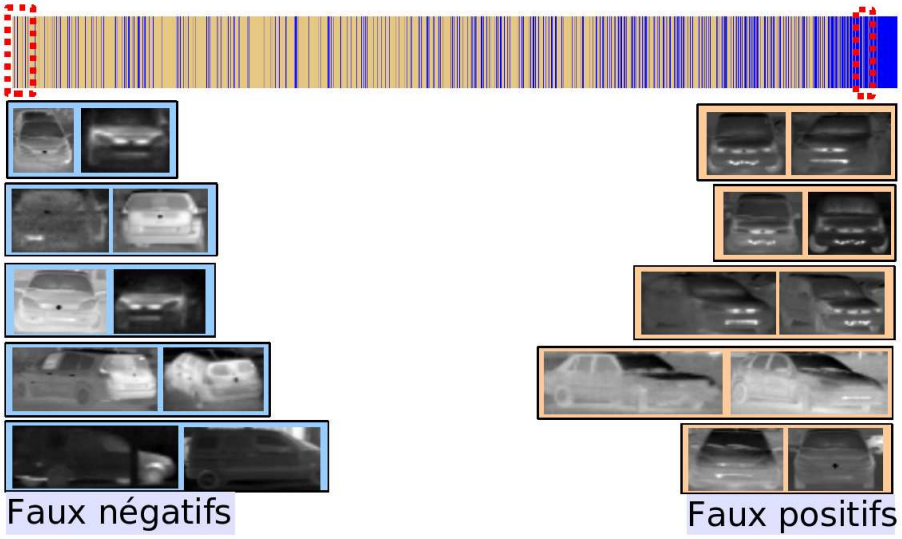


Figure 5.19: Base Robin: Visualisation de faux positifs et de faux négatifs. Les images affichées correspondent aux zones en pointillé rouge. On constate que les faux positifs ont une apparence très similaires, et que les faux négatifs ont une apparence très différente, d'où la confusion de l'algorithme.

Bilan sur toutes les bases

L'analyse qualitative et quantitative de l'algorithme sur plusieurs bases de données de paires d'images identiques ou différentes nous permet de dresser un bilan de ses performances et de son comportement. L'algorithme gère correctement les difficultés liées aux occultations, aux petits changements d'orientation, à la présence de fond chargé. Il gère mal les grandes différences d'apparence visuelle: grande différence d'orientation, occultations importantes, variations thermiques importantes pour les images infra-rouge. Enfin, l'algorithme n'est pas très performant quand les paires d'apprentissage contiennent des paires positives d'apparence très différentes, car l'algorithme apprend à être tolérant à de grandes variations d'apparence, ce qui se généralise mal aux modèles inconnus.

5.4.7 Approches n'ayant pas abouti

Nous avons évalué différentes idées qui n'ont donné les résultats espérés. Nous les présentons ci-dessous, et analysons les causes de l'échec.

Classification par distance entre descripteurs d'images

Devant le succès de la représentation d'image par sac-de-mots, nous avons voulu évaluer la performance d'un algorithme qui prédit la similarité entre objets jamais vus à partir de leur

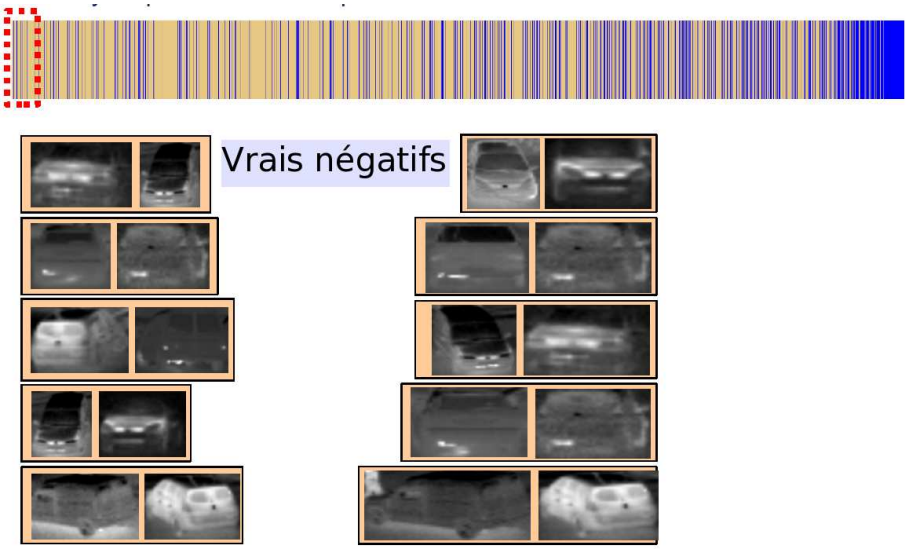


Figure 5.20: Base Robin: Visualisation de vrais négatifs. Les images affichées correspondent aux zones en pointillé rouge: grande confiance dans la prédiction "Différent".

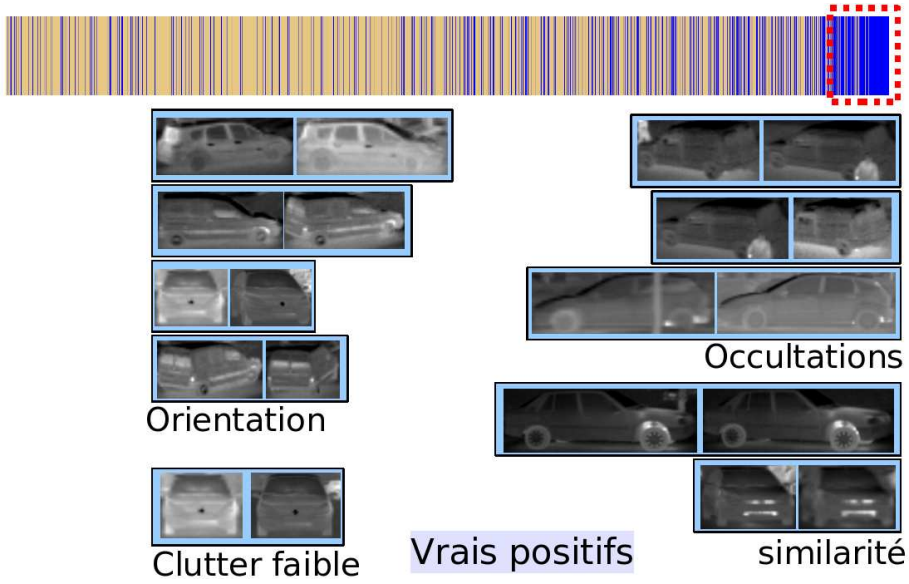


Figure 5.21: Base Robin: Visualisation de vrais positifs. Les images affichées correspondent aux zones en pointillé rouge: grande confiance dans la prédiction "Similaire".

représentation sac-de-mots. Sur la base Ferencz, nous avons donc calculé un vocabulaire visuel de 2000 éléments (mots visuels échantillonnés aléatoirement dans les images entre 15x15 et 60x60 pixels) et avons représenté les images par l'histogramme des occurrences des

mots visuels. Ensuite, nous avons représenté les paires d'images par un vecteur binaire. Pour chaque mot de vocabulaire visuel, on crée deux entrées dans le vecteur binaire. La première vaut un si le mot visuel apparaît dans au moins une des images de la paire, et zéro sinon. La seconde vaut un si le mot visuel apparaît dans les deux images, et zéro sinon. Ces entrées caractérisent et comparent les occurrences des mots du vocabulaire visuel dans les images. Un SVM est appris pour séparer les paires d'apprentissage positives et négatives, ce qui permet de prédire la similarité des paires d'images de test. Cette méthode permet d'obtenir une performance P-EER-PR de 50%, quand le hasard obtient 14% mais que la méthode proposée obtient 91.0% (section 5.4.4), ces chiffres sont rassemblés dans le tableau 5.10.

Cette faiblesse de la représentation par sac-de-mots s'explique aisément. Chaque image est approximée lors de la représentation (les vraies parties d'images sont représentées par les mots de vocabulaire visuel les plus proches), et on compare deux représentations approximatives pour évaluer la similarité. Hors, en particulier sur la base Ferencz, ce sont les petits détails (comme les poignées de porte) qui permettent de distinguer les véhicules. Au contraire, la méthode proposée considère les différences précises entre véhicules, et la quantification se fait sur cette différence (et non pas sur les parties d'images).

Descripteur de paire d'images: prise en compte de l'historique

Dans la méthode proposée, le vecteur descripteur x d'une paire d'images a autant de dimensions qu'il y a de feuilles dans la foret, et un "1" indique que la feuille a été atteinte par une paire de régions locales échantillonnée dans la paire d'image, un "0" indique le contraire.

Lors de la création de l'arbre, les noeuds sont subdivisés jusqu'à l'obtention de feuilles pures (ne contenant que des paires de régions locales positives ou négatives). Chaque noeud est donc à l'origine d'un sous-arbre. On pourrait imaginer que l'information importante est que la paire de régions locales ait atteint ce noeud, et qu'ensuite la feuille atteinte dans le sous-arbre issu de ce noeud importe peu. Les informations de structure de l'arbre n'étant pas codées dans le vecteur descripteur, le fait d'avoir atteint telle ou telle feuille du sous arbre ne permet pas de remonter à l'information que ces paires de régions locales sont passées par ce noeud qui est la racine du sous-arbre.

Nous avons donc proposé de prendre en compte cette information. Nous avons proposé un nouveau type de descripteur de paire d'images, qui a autant de dimensions qu'il y a de noeuds dans la foret. Quand une paire de régions locales échantillonnée dans une paire

Méthode	P-EER-PR
Chance	14%
Sac-de-mots	50%
Méthode proposée (cf section 5.4.4)	91.0±0.6

Table 5.10: Comparaison entre la méthode proposée et une représentation d'image par sac-de-mots

d'image atteint un noeud (que ce soit une feuille ou un noeud interne) la dimension correspondante est mise à un dans le vecteur descripteur. Si un noeud n'est jamais atteint, la dimension correspondante prend la valeur zéro.

Nous avons évalué cette représentation sur la base des voitures jouets. Nous utilisons 50 000 paires de régions locales positives et négatives pour l'apprentissage de l'arbre, choisissons la meilleure condition binaire parmi 500, une foret de 5 arbres, 150 paires de régions locales sont échantillonnées par paire d'image, nous échantillonnons les régions locales entre 30 et 100 pixels de hauteur, les conditions binaires sont basées sur des informations de niveau de gris. Il y a approximativement 30 000 noeuds par arbre, ce qui fait un vecteur descripteur de paire d'images de taille 150 000 approximativement (5 arbres de 30 000 noeuds). Avec cette nouvelle représentation, nous obtenons une performance (P-EER-PR) de 74%, c'est le même score qu'avec une représentation classique (sans les noeuds internes donc).

N'obtenant pas le gain de performance attendu, nous pouvons nous demander si la très haute dimensionalité du vecteur descripteur de paire d'images n'en est pas la cause. Nous avons donc effectué une réduction de dimensionalité par mesure information mutuelle conditionnelle [32]. Cette mesure évite de sélectionner des primitives trop redondantes. Les performances obtenues (P-EER-PR) sont présentées figure 5.22. On constate que la performance maximale est obtenue avec 10 000 primitives sélectionnées (soit moins de 10%), et que la hauteur moyenne des noeuds correspondants dans l'arbre est de 11.9 ± 4 au lieu de 17 ± 3 pour l'arbre complet. La réduction de dimension n'apporte qu'un gain de 1% à la performance totale, ce qui n'est pas significatif.

Ces résultats confirment les résultats obtenus section 5.4.3, qui indiquent que les arbres servent avant tout à fournir une quantification adaptée aux données manipulées, mais que leur calcul n'est pas crucial.

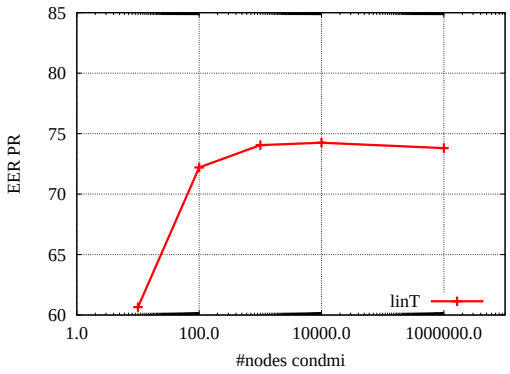


Figure 5.22: Performance (Précision au taux d'erreur égale de la courbe Précision-Rappel) en fonction du nombre de primitives sélectionnées par taux d'information mutuelle conditionnelle.

Extension aux catégories d'objets

Nous avons voulu étendre notre méthode aux catégories d'objets plutôt qu'aux instances d'objets. Dans cette section, les paires d'images similaires sont donc des paires d'objets de catégories identiques (mais les instances d'objets sont différents) et les paires d'images différentes sont des paires d'images de catégories différentes. Nous utilisons pour nos expérimentations la base Xerox7, qui contient 7 catégories d'objets. Nous sélections aléatoirement 30 images par catégorie pour l'apprentissage, à partir desquelles on crée 100 paires positives par catégorie pour l'apprentissage. Il y a autant de paires négatives que de paires positives, et les quantités sont identiques pour l'apprentissage et le test. Ces paires sont illustrées figure 5.23.

La méthode a été développée pour comparer des paires d'images du même objet ayant subi une faible transformation visuelle: faible changement d'orientation, faibles occultations, faible changement d'illumination, etc. Ainsi, quand on sélectionne une partie au hasard dans la première image, on peut s'attendre à trouver une partie similaire à une position proche dans la seconde image. Comme on le constate sur les images de la figure 5.23, cette hypothèse n'est plus valide dans le cadre des catégories d'objets où les objets ne sont pas (a) identiques (b) grossièrement alignés.

Il faut donc considérer d'autres méthodes de calcul des paires de régions locales. La première méthode considérée consiste à détecter des points d'intérêt (Harris-Laplace), à calculer un descripteur SIFT autour de ces points, et à faire des paires de régions locales en considérant toutes les paires dont la distance entre les descripteurs SIFT est inférieure à un seuil. Cette méthode donne une performance de 59% quand la chance est de 50% (voir tableau 5.11), ce qui est donc mauvais.

Pour obtenir un score de référence à dépasser, nous considérons des paires fabriquées totalement au hasard, c'est à dire que les deux descripteurs sont échantillonnés totalement au hasard dans chaque image. Cette méthode obtient un score de 58%. Cela signifie que les paires de régions locales obtenues par détecteurs de points d'intérêt sont très mauvaises, car ne sont pas significativement meilleures que des paires fabriquées au hasard. On peut constater sur la figure 5.24 que les paires de régions locales produites ne sont pas sémantiquement correctes.

Méthode	P-EER-PR
Chance	50%
Arbres + HL + toutes paires $\leq$ seuil	59%
Arbres + paires hasard total	58%
Classification par CB	63%

Table 5.11: Performance (précision lors du taux d'erreur égale de la courbe précision-rappel) du calcul de similarité sur des catégories d'objets (et non plus des instances). Aucune des méthodes ne se comporte significativement mieux que le hasard.

Figure 5.23: Exemples de paires d'apprentissage positives (lignes 1 à 4) et négatives (lignes 5 et 6) de la base Xerox7. Il s'agit dans cette expérimentation de comparer des catégories d'objet, et non plus des instances d'objet.

Enfin, nous avons mesuré la performance d'une représentation par sac-de-mots comme indiqué section 5.4.7, cette représentation donne une performance de 63%, ce qui est meilleur que la méthode proposée mais tout de même très faible.

On en conclut que la création de paires de régions locales correctes est cruciale pour la performance de l'algorithme, et il n'est pas possible d'obtenir de telles paires pour des catégories d'objets en considérant une distance euclidienne entre descripteurs génériques (niveaux de gris, SIFT) en raison de la grande variabilité intra-classe.

5.5 Discussion et conclusion

Nous nous sommes intéressés à la mesure de similarité entre deux images d'objets qui n'ont jamais été vus lors d'une phase d'apprentissage, mesure apprise à partir d'un ensemble d'apprentissage de paires d'images labélisées "identiques" ou "différentes". Nous avons proposé une solution innovante consistant à (a) produire des paires de régions locales correspondantes (b) quantifier ces paires de régions locales avec un ensemble d'arbres extrêmement aléatoires et (c) combiner les différentes quantifications pour prendre une décision globale sur la paire d'images.

Notre algorithme sélectionne et combine automatiquement différentes caractéristiques mesurées dans les images (géométrie et descripteurs SIFT). Nous avons montré que la mesure de similarité apprise n'est pas générique, mais au contraire très spécifique à la base de données sur laquelle les différents éléments sont appris (arbre, poids).

Nos expérimentations montrent que notre approche donne de très bons résultats sur les quatre bases de référence utilisées pour l'évaluation. Nous avons dépassé très significativement les résultats présentés dans l'état de l'art par Ferencz [24], Vidit [42] et Fleuret [31], et avons obtenu une haute performance sur notre propre base de données.

Nous avons considéré l'extension de ces travaux à des catégories d'objet, mais l'impossibilité d'obtenir de bonnes paires de régions locales est responsable de la très faible performance obtenue.

La base de voitures jouets, des exécutables de notre algorithme et d'autres informations sont disponibles à l'adresse suivante: http://lear.inrialpes.fr/people/nowak.

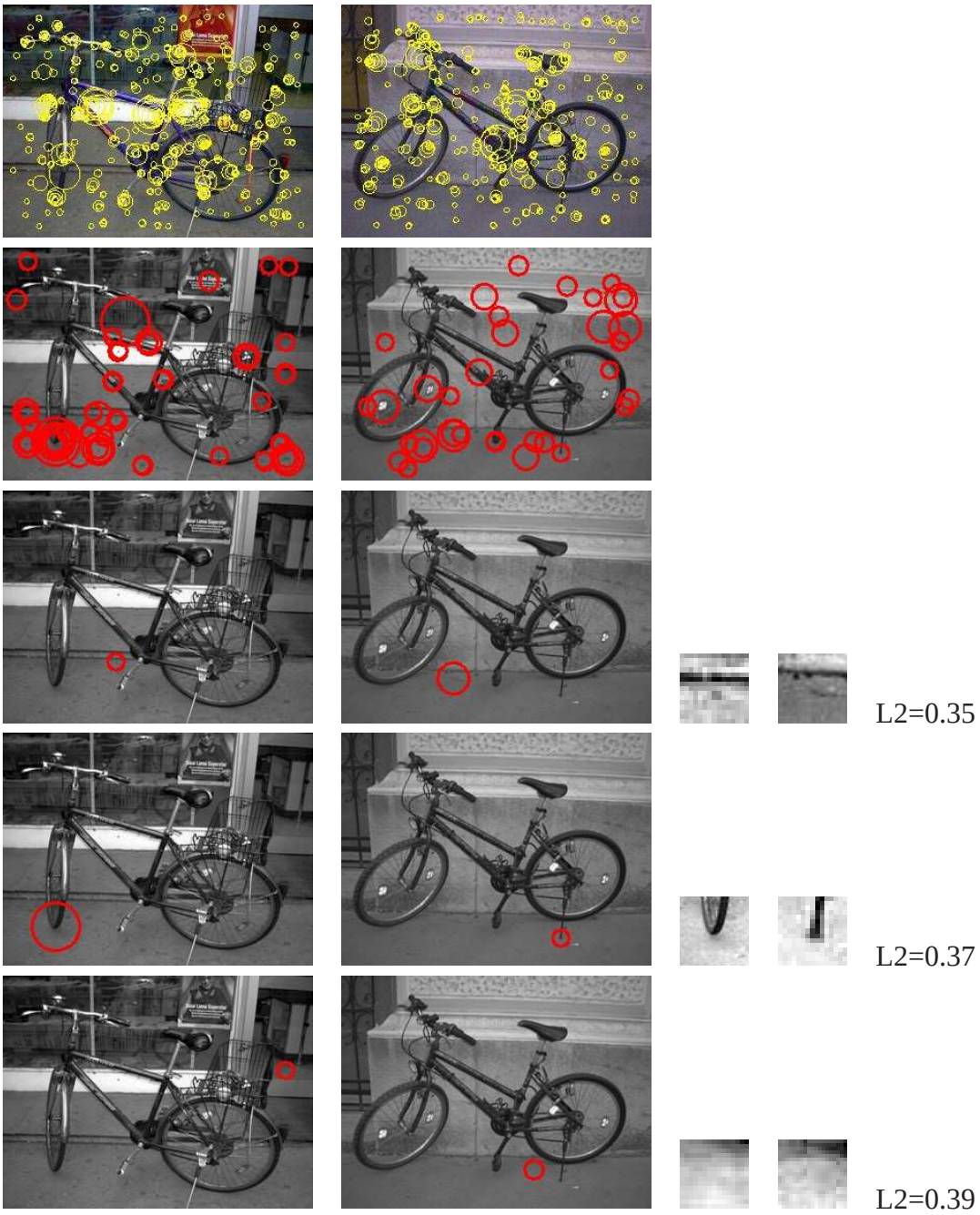


Figure 5.24: Visualisation des paires de régions locales construites. Ligne 1: les points d'intérêt détectés sur chaque image. Ligne 2: les points d'intérêt apparaissant dans des paires de régions locales. Lignes suivantes: les trois paires de régions locales (colonnes 3,4) dont la distance entre les descripteurs SIFT (colonne 5) est la plus faible. Ces paires ne sont clairement pas satisfaisantes.

SIXTH CHAPTER

Conclusion

Cette thèse s'est penchée sur la reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentations locales d'images. Il s'agit de proposer un algorithme qui prend une image en entrée, et qui donne la catégorie de l'objet contenu dans celle-ci en sortie. C'est un problème très difficile en raison des apparences multiples que peut prendre le même objet (problématique de reconnaissance *d'instances* d'objets) en raison de modifications de point de vue, d'illumination, de déformations, de la présence de fond chargé et d'occultations. Dans le cas de la reconnaissance de *catégories* d'objets, il faut ajouter à ces difficultés les variations intra-classe qui peuvent être importantes, il suffit pour s'en convaincre de considérer la diversité des différents modèles de voitures.

Diverses approches ont été considérées dans l'histoire de la vision par ordinateur pour palier ces difficultés. Les premières approches sont des méthodes globales qui modélisent l'image dans son ensemble, et presentent donc peu de robustesse aux modifications locales. Des méthodes locales ont ensuite été proposées pour gagner en robustesse, celles-ci considèrent les images comme des ensembles de parties locales, avec des relations géométriques entre ces parties plus ou moins contraintes. Nous nous sommes intéressés aux méthodes dites "sac-de-mots", qui ignorent totalement les relations géométriques entre les différentes parties. En dépit de la simplicité de ces métohdes, celles-ci obtiennent les meilleurs résultats de l'état de l'art. En effet, leur simplicité réduit les risques de surapprentissage et les rend très flexibles.

Nous présentons ci-dessous les principales contributions de cette thèse axée sur les représentations locales d'images pour la reconnaissance de catégories d'objets et d'instances d'objets, puis nous détaillons notre démarche et les principaux résultats obtenus. Enfin, nous mentionnons des perspectives d'évolution de nos travaux.

6.1 Principales contributions

Nos principales contributions pour la reconnaissance de catégories d'objets sont les suivantes.

- Nous avons montré l'influence relative des différents composants de la description sac-de-mots, et avons montré que le facteur principal est la quantité de régions locales sélectionnées (ou échantillonnées). De ce fait nous avons proposé un échantillonnage aléatoire d'une grande quantité de régions locales au lieu des détecteurs de points d'intérêt.

- Nous avons confirmé cette étude sur les images infra-rouges de véhicules (basse résolution, bruitées, multiclasses) et avons déterminé les principales difficultés opérationnelles: les occultations et la présence de fond chargé lorsque les cibles sont mal séparées du fond.

- Nous nous sommes intéressés au compromis performance / temps de calcul, et avons proposé un algorithme optimisant ce compromis. Il consiste à effectuer une sélection de primitives dans un classifieur hiérarchique, ce qui est plus efficace que la traditionnelle sélection de primitives pour classifeurs plats.

Notre principale contribution pour la reconnaissance d'instances d'objets est la suivante.

- Nous avons proposé une méthode de calcul de similarité entre objets jamais vus lors d'une phase d'apprentissage, ce qui permet de déterminer si les objets sont identiques ou différents. Il s'agit de calculer un score à partir de la quantification par une foret d'arbres extrêmement aléatoires de régions locales correspondantes échantillonnées dans les deux images à comparer.

6.2 Démarche et rappel des résultats

Dans cette section, nous faisons le point sur notre démarche et nous rappelons les résultats obtenus

6.2.1 Intérêt pour la méthode sac-de-mots

En raison du caractère industriel de cette thèse, nous avons travaillé dès le début sur des bases de données de véhicules en imagerie infra-rouge. Ces bases sont caractérisées par une faible résolution, des cibles de petite taille, un bruit important, des occultations très importantes par le relief et le paysage, des grandes variations des conditions thermiques des véhicules (se traduisant par des points chauds dans les images), ainsi que d'autres sources de variabilité plus classiques: changement de point de vue, d'illumination, etc.

Les images étant de petite taille, nous avons d'abord considéré les méthodes de représentation globales: pixels ayant subi une anayse en composante principale, filtrage par dérivées de gaussiennes, etc. Comme on pouvait s'y attendre, ces méthodes globales n'ont pas donné de bons résultats en raison de leur grand manque de robustesse.

Nous nous sommes alors intéressés aux méthodes locales plus modernes, telles que le modèle de forme implicite de Bastian Leibe et al [49], la méthode à géométrie encodée dans

le vecteur descripteur de Agarwal et al [2]. Ces méthodes étant basées sur des détecteurs de points d'intérêt et sur la détermination assez précise de la position des régions locales dans l'objet, elles n'ont pas donné de bons résultats avec nos images. En effet, celles-ci sont bruitées et de très faible résolution, et ne sont donc pas adaptées aux détecteurs de points d'intérêt.

Nous avons donc considéré des méthodes pouvant s'affranchir de la position des régions locales dans les images, et nous nous sommes naturellement orientés vers les métohdes sac-de-mots. Nous avons donc mené une étude poussée de cet algorithme.

6.2.2 Etude des algorithmes sac-de-mots

Le chapitre 2 étudie de manière approfondie les différents composants de l'algorithme sac-de-mots, et détermine les choix optimaux pour obtenir la performance maximale. Nous avons travaillé sur trois bases de texture et trois bases de catégories d'objets pour mener cette étude, et afin de valider nos conclusions nous avons comparé notre algorithme aux meilleurs algorithmes évalués sur la base de référence PASCAL VOC 2005, sur laquelle nous avons obtenu la meilleure performance.

Nous avons montré que le facteur le plus important pour la performance est la quantité de régions locales échantillonnées dans les images à décrire. De ce fait nous avons proposé d'échantillonner aléatoirement une grande quantité de régions locales. La méthode classique consiste à utiliser des détecteurs de points d'intérêt, mais ceux ci ne peuvent retourner qu'un nombre limité de points, et l'échantillonnage aléatoire (qui peut produire une quantité infinie de régions) se montre donc plus performant au final.

Nous avons aussi tiré des conclusions sur les autres composants de la méthode. Le vocabulaire visuel doit être de grande taille, et il doit être calculé à partir de régions similaires à celles échantillonnées dans les images: même méthode d'échantillonnage, même type d'image. Nous avons proposé une normalisation des histogrammes décrivant les images par maximisation d'un critère d'information mutuelle entre les primitives et les catégories. Nous avons aussi étudié l'effet du choix d'échelle minimale d'échantillonnage, où se joue le compromis entre régions informatives et bruit.

6.2.3 Algorithmes sac-de-mots et images infra-rouges

Etant donné notre objectif industriel, nous avons évalué l'intérêt de ces méthodes sur les images infra-rouges qui nous intéressent, c'est l'objet du chapitre 3. Nous avons obtenu de très bonnes performances sur les bases étudiées. Nous avons montré que les paramètres responsables de la performance sont les mêmes en imagerie visible et en imagerie infra-rouge.

L'une des préoccupations majeures du partenaire industriel était la quantité de paramètres sensibles à régler pour obtenir une bonne performance. Nous avons donc analysé la sensibilité de tous les paramètres, et avons montré qu'il n'y a aucun paramètre à régler manuellement. En effet, l'étude paramétrique montre que les valeurs peuvent être prédites facilement

en observant les performances sur les données d'apprentissage (cas de la taille du vocabulaire) ou bien que certaines valeurs doivent être réglées de manière à avoir la finesse optimale (cas de l'offset pour l'échantillonnage).

Grâce aux données hybrides qui peuvent être générées à profusion, nous avons pu mener la première étude de performance en fonction des paramètres opérationnels, tels que la distance cible-caméra, le contraste interne des cibles, le contraste du fond, la luminosité moyenne du fond et des cibles, la transmission atmosphérique, la précision du détourage des cibles, le taux d'occultation des véhicules, le nombre de classes à reconnaître, etc. Nous avons montré que les deux paramètres opérationnels les plus sensibles sont le taux d'occultation et la présence de fond chargé quand les cibles sont mal ségementées du fond.

6.2.4 Compromis temps de calcul / performance

Les algorithmes développés étant susceptibles d'être implantés en hardware pour une utilisation sur le terrain, nous nous sommes aussi intéressés au compromis entre temps de calcul et performance, car sur le terrain l'un ne va pas sans l'autre. C'est l'objet du chapitre 4.

Le facteur le plus limitant dans notre chaîne algorithmique étant la détection des régions locales, nous avons naturellement investigué les téchniques de sélection de primitives, pour réduire la quantité de primitives à détecter dans les images. Nous avons constaté que la selection de primitives classique pour classifieurs multiclasses est sous optimale, et nous avons proposé une méthode de sélection de primitives pour classifieurs hiérarchiques, beaucoup plus efficace. Celle-ci tire parti de l'élimination rapide de la majorité des classes, et donc les primitives sélectionnées deviennent de plus en plus spécialisées, ce qui fait qu'un nombre plus faible d'entre elles est requis.

6.2.5 Identification d'objets jamais vus

Enfin, nous nous sommes intéressés à la plus grande préoccupation de nos partenaires industriels: comment traiter des objets qui n'ont jamais été vus lors de l'apprentissage? Car ce n'est pas tout de bien reconnaître les objets qui ont été appris, il faut aussi pouvoir réagir quand de nouveaux objets sont vus.

Si deux objets n'ont jamais été vus lors de l'apprentissage, alors il n'est pas possible de déterminer leur classe: celle-ci est inconnue. Les méthodes présentées ci-dessus ne nous sont donc d'aucune utilité. Par contre, il est possible de dire si ces deux objets sont identiques ou différents. Ce qui revient à identifier des objets jamais vus lors de l'apprentissage. Cette étude est l'objet du chapitre 5.

Nous avons donc proposé un algorithme qui donne un score de similarité entre deux objets jamais vus lors de l'apprentissage. Les données d'apprentissage sont des paires d'images, labélisées "similaires" quand les objets contenus dans ces images sont identiques, et labélisées "différentes" dans le cas contraire. Nous utilisons toujours une représentation locales, mais celle-ci ne concerne plus les images mais les paires d'images. Nous échantillonnons des paires de régions correspondantes dans les deux images, et nous les quantifions

à l'aide de forêts d'arbres extrêmement aléatoires. La paire d'image est alors représentée par l'histogramme d'occurrences des feuilles des arbres.

Cet algorithme novateur a été évalué sur notre propre base de voitures jouets, et nous avons obtenu de hautes performances. Nous l'avons aussi évalué sur une base du projet Robin, qui contient des véhicules en imagerie infra-rouge. Malgré toutes les difficultés spécifiques à l'infra-rouge (mentionnées plus haut) nous avons obtenu des résultats très intéressants, qui sont aujourd'hui analysés pour investiguer de nouvelles pistes. Nous avons comparé notre algorithme aux meilleurs algorithmes de l'état de l'art, qui ont été réalisés spécifiquement pour des bases de voitures, de visages, et d'objets hétéroclites, et notre algorithme les a tous très significativement dépassés.

6.3 Perspectives

Nos travaux sont loin d'être un aboutissement, et de nombreuses pistes peuvent être investiguées pour les améliorer.

Nous pensons tout d'abord à la classification rapide et fiable d'images, ce qui était l'objet du chapitre 4. Nous avons considéré l'algorithme de Moosmann et al [63] pour la classification d'images. C'est l'algorithme à base de forêts d'arbres extrêmement aléatoires que nous avons utilisé pour les paires d'images, mais cette fois ci celui ci est utilisé pour les images simples, et donc les données qui passent par les feuilles sont des régions locales, et non plus des paires de régions locales. Cet algorithme obtient un très bon compromis temps / performance. Nous avons donc poussé les recherches un peu plus loin, et avons utilisé cet algorithme pour la localisation d'objets dans des images, inspirés en particulier par [64]. L'algorithme établit une carte de saillance à partir de l'image, c'est à dire une carte des régions probables où se trouve l'objet. Cette carte est obtenue par analyse du score obtenu par les régions locales qui traversent les arbres de la foret. Une région locale qui obtient un vote "objet" par tous les arbres de la foret générera un score de confiance élevé à la position où cette région locale a été échantillonnée, et inversement. Nous avons poussé l'étude menée par Moosmann, et avons déterminé des facteurs influents sur la localisation, comme l'utilisation de masques dans les images d'apprentissage. Cependant, les facteurs que nous avons identifié ne sont pas suffisants pour obtenir une localisation satisfaisante, et il faudrait donc poursuivre des travaux dans cette direction. La piste la plus sérieuse que nous envisageons est de déterminer les régions locales les plus fiables pour la localisation, puis d'uiliser des contraintes géométriques sur les paires de régions locales pour obtenir une localisation précise. Les paires de régions sont plus robustes et plus simples à utiliser que des modèles en étoile ou des modèles géométriques rigides.

Ensuite, nous pouvons penser à étendre les travaux menés dans le chapitre 5 sur la classification de paires d'images. Nous avons obtenu de très bons résultats sur la classification de paires d'instances d'objets, et nous souhaitons maintenant travailler sur la classification de paires d'images toutes orientations confondues, tous modèles confondus. Nous avons vu que c'est un problème très difficile, car il n'est pas possible d'obtenir par une simple mesure de

distance des paires de régions locales correspondantes. En effet, les apparences visuelles des régions locales varient beaucoup quand les modèles et les conditions d'observations varient d'une vue à l'autre. Nous pensons donc utiliser une autre mesure qu'une distance L2 et une correlation pour déterminer la région locale de la seconde image qui correspond le mieux à celle échantillonnée dans la première image. La mesure de distance serait donnée par une forêt d'arbres extrêmement aléatoires, qui seraient appris à partir de photos calibrées, ce qui permet de déterminer les régions correspondantes précises de chaque région locale de la première image. Cela résoudrait la principale difficulté que nous avons rencontrée.

Bibliography

[1] A. Agarwal and B. Triggs. Hyperfeatures – multilevel local coding for visual recognition. In *ECCV*, May 2006. 35, 37

[2] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, November 2004. 35, 37, 43, 98, 100, 175

[3] S. Agarwal and D Roth. Learning a sparse representation for object detection. In *ECCV*, pages 113–130, 2002. 29

[4] A. Bar Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, (6):937–965, 2005. 139

[5] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, pages 26–33, 2005. 30

[6] T Berg, A Berg, J Edwards, M Maire, R White, Y Whye Teh, E Learned-Miller, and Forsyth. D. Names and faces in the news. *CVPR*, pages 848–854, 2004. 147

[7] I. Biederman. *An Invitation to Cognitive Science, Vol. 2: Visual Cognition*. MIT press, 1995. 17

[8] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 22, 23

[9] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, volume 1, pages 710–715, June 2005. 35

[10] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. 101

[11] Leo Breiman. Random forests. *ML Journal*, 45(1):5–32, 2001. 144

[12] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV*, pages 628–641, 1998. 29

[13] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR '05*, pages 539–546, 2005. 139

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991. 104

[15] T.M. Covert and P.E. Hart. Nearest neighbour pattern classification. *Transactions on Information Theory*, pages 21–27, 1967. 101

[16] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004. 28, 30, 34, 35, 36, 37, 43, 45, 98

[17] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 21, 27, 28

[18] Sanmay Das. Filters, wrappers and boosting-based hybrid for feature selection. In *ICML*, 2001. 103

[19] M. Everingham *et al*. The 2005 pascal visual object classes challenge. In Springer-Verlag, editor, *First PASCAL Challenges Workshop*, 2006. 29, 48, 50

[20] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf. 29

[21] M.R. Everingham, J. Sivic, and A. Zisserman. 'hello! my name is... buffy' - automatic naming of characters in tv video. In *BMCV*, pages 889–908, Septembre 2006. 152

[22] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, April 2006. 29, 139

[23] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 30

[24] A. Ferencz, E.G. Learned Miller, and J. Malik. Building a classification cascade for visual identification from one example. In *ICCV'05*, pages I: 286–293, 2005. 139, 155, 171

[25] A.D. Ferencz, E.G. Learned-Miller, and J. Malik. Learning hyper-features for visual identification. In *NIPS'05*, pages 425–432. 2005. 139, 140, 141, 147

[26] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, pages II: 1816–1823, 2005. 35, 36, 37, 98

[27] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR03*, pages II: 264–271, 2003. 29, 34, 35, 37, 43

[28] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Transactions on Computer*, 22(1):67–92, 1973. 29, 30

[29] M.A. Fischler and R.C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *ACM*, 24(6):381–395, 1981. 24, 25

[30] A.W. Fitzgibbon and A. Zisserman. Joint manifold distance: a new approach to appearance based clustering. In *CVPR'03*, pages I: 26–33, 2003. 139

[31] F. Fleuret and G. Blanchard. Pattern recognition from one example by chopping. In *NIPS'05*, pages 371–378. MIT Press, 2005. 139, 147, 155, 156, 171

[32] Francois Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, pages 1531–1555, 2004. 103, 168

[33] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, pages 1289–1305, 2003. 103

[34] Y. Freund and E. Shapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 5:119–139, 1997. 28

[35] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006. 98, 144

[36] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS'05*, 2005. 139

[37] C Goad. Special purpose automatic programming for 3d model-based vision. In *Proceedings of the DARPA Image Understanding Workshop*, pages 371–381, 1983. 24, 25

[38] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS'04*, 2004. 139

[39] K. Grauman and T.J. Darrell. Efficient image matching with distributions of local invariant features. In *CVPR05*, pages II: 627–634, 2005. 35, 37, 43

[40] W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *PAMI*, 9(4):469–482, 1987. 24, 25

[41] D.P. Huttenlocher and S Ullman. Recognizing solid objects by alignment with an image. *IJCV*, 5(2):195–212, 1990. 24, 25

[42] V. Jain, A. Ferencz, and E.G. Learned Miller. Discriminative training of hyper-feature models for object identification. In *BMVC*, pages I, pp. 357–366, 2006. 147, 152, 155, 156, 171

[43] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *ECML-98, 10th European Conference on Machine Learning*, pages 137–142. Springer Verlag, 1998. 35

[44] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *ICML*, pages 121–129, 1994. 103

[45] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, 2005. 35, 43, 98, 100, 101, 108, 121

[46] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Object recognition by affine invariant matching. In *CVPR*, pages 335–344, 1988. 24, 25

[47] S. Landeau and T. Dagobert. Image database generation using image metric constraints: an application within the caladiom project. In *SPIE Volume 6234*, 2006. 57, 58

[48] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *ICCV*, pages 649–655, 2003. 36, 43

[49] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003. 29, 35, 37, 43, 98, 100, 174

[50] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, June 2001. 34, 36, 37, 43, 100

[51] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *IJCV*, 11(3):283–318, December 1993. 43

[52] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *Transactions on Knowledge and Data Engineering*, pages 491–502, 2005. 103

[53] D. Lowe. Object recognition from local scale-invariant features. In *CVPR*, pages 1150–1157, 1999. 26

[54] D.G. Lowe. The viewpoint consistency constraint. *IJCV*, 1(1):57–72, 1987. 20, 24, 25

[55] D.G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural computation*, 7(1):72–85, 1995. 139

[56] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004. 35, 36, 37, 39, 43, 139, 145, 150

[57] E. Haritaglu M. Betke and L. Davis. Multiple vehicle detection and tracking in hard real time. In *IEEE Intelligent Vehicles Symposium*, 1996. 98

[58] V. N. mapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995. 101

[59] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV02*, page I: 128 ff., 2002. 35, 36, 37, 43

[60] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Computer Vision*, 65(1/2):43–72, 2005. 43

[61] E.G. Miller, N.E. Matsakis, and P.A. Viola. Learning from one example through shared densities on transforms. In *CVPR'00*, pages I: 464–471, 2000. 139

[62] D. Mladenic and M. Grobelnik. Feature selection on hierarchy of web documents. *Decis. Support Syst.*, 35(1):45–87, 2003. 103, 104, 118

[63] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *PAMI*, to appear, 2007. 31, 177

[64] F. Moosmann, B. Triggs, and F. Jurie. Randomized clustering forests for building fast and discriminative visual vocabularies. In *NIPS'06*. 2006. 98, 141, 144, 145, 177

[65] H Murase and S Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14(1):5–24, 1995. 25, 27

[66] S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, CUCS, 1996. 147

[67] W. Niblack, R. Barber, W. Equitz, M.D. Flickner, D. Glasman, D. Petkovic, and P. Yanker. The qbic project: Querying image by content using color, texture, and shape. *SPIE*, 1908:173–187, February 1993. 25, 27, 36

[68] E. Nowak and F. Jurie. Vehicle categorization: Parts for speed and accuracy. In *VS-PETS*, 2005. 31, 47, 98

[69] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *CVPR*, pages 1–8, 2007. 31

[70] Eric Nowak, Frederic Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*. Springer, 2006. 30, 98, 141, 143, 149

[71] A. Opelt, A. Fussenegger, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004. 28

[72] Platon. *La République, Livre VII*. IV av JC. 17, 23

[73] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *PAMI*, 20(6):637–646, 1998. 25, 27

[74] S. Rajan and J. Ghosh. *An Empirical Comparison of Hierarchical vs. Two-Level Approaches to Multiclass Problems*. Springer, 2004. 102, 109, 117, 118, 121, 134

[75] Alain Rakotomamonjy and Frédéric Suard. Svm variable selection with application to pedestrian detection. In *RFIA*, 2004. 103, 105

[76] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *CVPR*, pages 271–278, 2005. 30

[77] I. Rish. An empirical study of the naive bayes classifier. In *Workshop on Empirical Methods in Artificial Intelligence*, 2001. 101

[78] Monica Rogati and Yiming Yang. High-performing feature selection for text classification. In *CIKM*, 2002. 103

[79] B. Schiele and J. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 36(1):31–50, 2000. 25, 27

[80] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *PAMI*, 19(5):530–534, 1997. 26

[81] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, pages 746–751, 2000. 27

[82] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *ICML '04*, 2004. 139

[83] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV03*, pages 1470–1477, 2003. 35, 37, 43, 98

[84] G Stockman. Object recognition and localization via pose clustering. In *Computer Vision, Graphics and Image Processing*, pages 361–387, 1987. 24

[85] D.W. Thompson and J.L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *International Conference on Robotics and Automation*, pages 208–220, 1987. 24

[86] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR04*, pages II: 762–769, 2004. 27, 28, 98, 100

[87] M. Turk and A. Pentland. Face recognition using eigenfaces. In *CVPR*, pages 586–591, 1991. 25, 27

[88] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998. 23, 25, 28, 66, 101

[89] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV03*, pages 281–288, 2003. 100

[90] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001. 27, 28, 65

[91] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, pages I: 18–32, 2000. 29, 35, 37, 43

[92] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS'05*. 2006. 139

[93] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *International Workshop on Learning for Adaptable Visual Systems (LAVS04), Cambridge, United Kingdom*, 2004. 100

[94] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV*, 2005. 35, 36, 37, 43, 46, 98

[95] H.J. Wolfson and I Rigoutsos. Geometric hashing: An overview. *Computational Science and Engineering*, 4(4):10–21, 1997. 24, 25

[96] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS'02*, 2002. 139

[97] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997. 103, 104

[98] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classifcation of texture and object categories: An in-depth study. Technical Report RR-5737, INRIA Rhône-Alpes, Nov 2005. 28, 49, 50, 98